

Heitor Medrado de Faria



BACULA

0 software livre de backup



VISIT...

LANZAROTE
Caliente.COM

Heitor Medrado de Faria



BACULA

O software livre de backup



Heitor Medrado de Faria

BACULA

O software livre de backup

3ª Edição



Todos os direitos reservados. Nenhuma parte deste livro poderá ser reproduzida, sob qualquer meio, especialmente em fotocópia (xerox), sem a permissão, por escrito, da Editora.

Editor: Sergio Martins de Oliveira

Diretora: Rosa Maria Oliveira de Queiroz

Gerente de Produção Editorial: Marina dos Anjos Martins de Oliveira

Editoração Eletrônica: Abreu's System

Capa: Use Design

Desenvolvimento de eBook: Loope – design e publicações digitais | www.loope.com.br

Técnica e muita atenção foram empregadas na produção deste livro. Porém, erros de digitação e/ou impressão podem ocorrer. Qualquer dúvida, inclusive de conceito, solicitamos enviar mensagem para editorial@brasport.com.br, para que nossa equipe, juntamente com o autor, possa esclarecer. A Brasport e o(s) autor(es) não assumem qualquer responsabilidade por eventuais danos ou perdas a pessoas ou bens, originados do uso deste livro.

BRASPORT Livros e Multimídia Ltda.

Rua Pardal Mallet, 23 – Tijuca

20270-280 Rio de Janeiro-RJ

Tels. Fax: (21)2568.1415/2568.1507

e-mails: marketing@brasport.com.br

vendas@brasport.com.br

editorial@brasport.com.br

www.brasport.com.br

Filial SP

Av. Paulista, 807 – conj. 915

São Paulo-SP

Agradecimentos

*A Deus e aos meus
mentores de software livre:
Julio Neves e Kern Sibbald.*

*À minha esposa,
à minha schnauzer Sienna,
à minha família e
à comunidade nacional de
usuários do Bacula.*

Sumário

1 | Conceitos de Backup

- 1.1 Princípios do backup
- 1.2 O futuro do backup
- 1.3 Mitos do backup
- 1.4 Glossário
- 1.5 Níveis de backup
- 1.6 Mídias de backup
- 1.7 Local de armazenamento do backup

2 | Passos para Escolha e Implantação de um Sistema de Backup

- 2.1 Existe uma política de backup?
- 2.2 Quais são os serviços e negócios backupeados?
- 2.3 Quais aplicações específicas e sistemas operacionais precisam de backup?
- 2.4 Qual o tamanho do backup?
- 2.5 Qual é a capacidade de armazenamento atual?
- 2.6 O software de backup tem compatibilidade com o armazenamento?
- 2.7 Qual formato de armazenamento utilizar para o backup?
- 2.8 Existe algum legado que precise ser tratado?
- 2.9 O que mais considerar?

3 | Política de Backup e Melhores Práticas

- 3.1 Política de backup
- 3.2 Melhores práticas do backup

4 | Esquema GFS de Rotação de Backups

- 4.1 Backups escalonados
- 4.2 Detalhes do GFS
- 4.3 Exemplos de GFS

5 | O Que é o Bacula?

- 5.1 Módulos do Bacula
- 5.2 Funcionalidades únicas do Bacula
- 5.3 Principais funcionalidades do Bacula

6 | Instalação do Bacula

- 6.1 Instalação do servidor – banco de dados de catálogo
- 6.2 Instalação do servidor – daemons do Bacula e dependências
- 6.3 Construindo o servidor Bacula a partir do código-fonte (compilação)
- 6.4 Instalação do Bacula e resolução de problemas no início de daemons

7 | Configuração do Bacula

- 7.1 Esboço da primeira instalação do Bacula
- 7.2 bacula-dir.conf
- 7.3 bacula-sd.conf
- 7.4 bacula-fd.conf
- 7.5 bconsole.conf (bat.conf)
- 7.6 Novos clientes Bacula
- 7.7 Envio de e-mail do Bacula
- 7.8 ACL de consoles (contas de usuário)
- 7.9 Usando includes para dividir arquivos de configuração do Bacula
- 7.10 Ciclo de vida dos volumes do Bacula
- 7.11 Compressão dos dados do backup
- 7.12 Jobs de cópia e migração
- 7.13 Etiquetagem automática de volumes
- 7.14 Deduplicação em nível de arquivos
- 7.15 Criptografia do tráfego das comunicações do Bacula (TLS)
- 7.16 Informações de diretivas e configuração TLS
- 7.17 Criptografia dos dados de backup
- 7.18 Pool Scratch
- 7.19 Clientes de backup em redes diferentes

8 | Dispositivos de Armazenamento em Disco

- 8.1 Limitando o tamanho do volume de disco

9 | Backup Bacula em Fitas Magnéticas

- 9.1 Dispositivos de robôs de fitas
- 9.2 Outras operações de fitas
- 9.3 Manipulação de robôs de fitas
- 9.4 Escrever jobs de backup em vários drives simultaneamente
- 9.5 Data spooling
- 9.6 Imprimir novas etiquetas de código de barras
- 9.7 Fitas de limpeza
- 9.8 Configuração e automação de drive de fita manual

10 | Fatores de Performance e Tuning

- 10.1 Fatores de desempenho do backup
- 10.2 Identificando gargalos

10.3 Tuning do servidor do Bacula

11 | Instalando Interfaces Gráficas

11.1 BAT – Ferramenta de administração Bacula (desktop)

11.2 Webacula 7.x

11.3 Webmin (permite alterar graficamente as configurações do Bacula)

11.4 Baculum: mais uma interface de configuração e administração

11.5 Bacula-web (interface de relatórios e estatísticas)

11.6 Outros projetos

12 | Comandos do Bacula

12.1 Comandos mais importantes do Bacula

12.2 Comandos situacionais do Bacula

12.3 Comandos especiais de redirecionamento (@)

12.4 Redirecionamento do shell para o bconsole

12.5 Comandos automáticos do bconsole durante o backup (e substituição de caracteres)

13 | Restauração Interativa pela Linha de Comando

13.1 O menu do comando restore

13.2 Árvore de seleção de arquivo para restauração da CLI

14 | Recuperação de Desastres do Servidor Bacula

14.1 Restauração do catálogo

14.2 bscan

14.3 bls e bextract

15 | Atualizando o Bacula

15.1 Etapas gerais de atualização

16 | Backup de Aplicações Específicas

16.1 O problema

16.2 Instalação e visão geral do plug-in Bacula bpipe

16.3 Exemplos de backup de aplicações com bpipe

16.4 Outras considerações sobre o backup de aplicações específicas e exemplos de scripts

17 | Procedimentos Específicos do Windows

17.1 Instalando o cliente Bacula e consoles

17.2 System State

17.3 Exemplos de FileSet de Windows

Referências

Anexo 1. Página de “Cola” do Bacula (bconsole)

Anexo 2. Códigos de Jobs do Bacula

Anexo 3. Expansão de Variáveis

Licenças do Bacula Enterprise, Treinamento e Suporte Profissionais do Bacula Community

1

Conceitos de Backup

Backup consiste em gerar dados redundantes com o propósito específico de recuperação no caso de perda dos originais.

As cópias podem ser armazenadas no mesmo computador, em um dispositivo de storage ou, ainda, em outro prédio ou localização (protegendo assim os dados de ameaças físicas e lógicas locais): partindo de um nível menor de segurança para um maior, respectivamente.

Independentemente de como os dados forem perdidos (apagamentos acidentais, corrupção dos dados etc.), um backup eficiente consiste naquele que minimiza os impactos dessa perda, possibilitando a restauração do arquivo ou serviço no menor tempo possível e com o mínimo de defasagem em termos de alteração das informações, aderente aos acordos operacionais ou de nível de serviço vigentes.

Os dispositivos de armazenamento mais comuns são robôs de fitas, discos rígidos, arranjos de discos, discos removíveis, backup externo em nuvem e fitas magnéticas manuais (mais dolorosas de manipular e raras atualmente).

Backups em nuvem são relativamente novos e resolvem muitos dos problemas de proteção das cópias de segurança, copiando os dados do prédio original para um ambiente de terceiros ou até mesmo um serviço de nuvem privada. No entanto, ainda é limitado pelas capacidades de transmissão das redes atuais.

As cópias de segurança podem sofrer influência da natureza dos dados, do número e do tamanho dos arquivos, das necessidades de negócio, da infraestrutura disponível, das funcionalidades do software de backup e das aplicações que eventualmente necessitem de artifícios especiais e ações pré-backup para uma cópia íntegra. Além disso, outros fatores devem ser observados: a janela de backup, a periodicidade necessária de agendamento dos jobs, o tempo que as cópias precisarão ser mantidas (retenção), a capacidade de armazenamento, o número de versões necessárias de cada dado backupeado e a compressão e eventual encriptação dos dados.

Quanto à topologia, os backups podem ser classificados entre centralizados ou descentralizados. Na modalidade centralizada, geralmente há um servidor de backup que comanda a realização de cópias de segurança, conferindo maior praticidade na administração e economicidade pelo armazenamento dos dados em um ou alguns dispositivos de armazenamento, com ganhos de escala.

Geralmente o backup é realizado pela noite, para que haja mínima possibilidade de impacto

1.1 Princípios do backup

Os processos de backup devem ter os seguintes fatores como guias gerais:

- Automação:** rotinas de backup devem ser executadas da maneira mais automatizada possível. Primeiro, porque a intervenção humana aumenta os riscos de erro de operação; segundo, porque alocar pessoas para algumas atividades simples, chatas e repetitivas é caro (por exemplo: ter uma pessoa apenas para efetuar troca de fitas magnéticas de drives manuais). Mesmo quando um robô de fitas é adquirido, é necessário verificar se está corretamente dimensionado, ou seja, se tem uma quantidade suficiente de gavetas (slots) para evitar, ao máximo, a remoção ou inserção manual de fitas no equipamento. De igual sorte, a reciclagem dos volumes (reutilização) é muito mais segura e menos trabalhosa se realizada automaticamente pelo algoritmo do software de backup de acordo com os parâmetros definidos, normalmente tempos de retenção.

- Controle de qualidade:** uma vez que o processo de backup deve ser automatizado na maior parte, é muito importante garantir que tudo funcione bem – desde a configuração do job até a execução do backup e da restauração. Novas rotinas devem ser sempre testadas; os logs de execução dos jobs devem ser sempre verificados; backups de dados e de aplicação devem ser testados periodicamente.

- Gestão de riscos:** como a maioria das disciplinas de segurança, é muito importante discutir, classificar e definir ações para mitigar os riscos mais perigosos na manutenção de dados corporativos e operações de backup, conforme detalhado no capítulo referente às políticas de backup.

- Aderência aos acordos operacionais ou de nível de serviço:** a mencionada tratativa de riscos e o nível de proteção provido pelo sistema de backup precisam estar alinhados às expectativas dos clientes, normalmente formalizados através dos acordos operacionais ou de nível de serviço. De igual sorte, é necessário determinar os componentes de custo e o orçamento disponível para os processos de backup, na medida em que os diferentes níveis de proteção contratados também implicam em custos menores ou maiores.

1.2 O futuro do backup¹

Os cenários atuais indicam um aumento na complexidade dos ambientes e no tamanho dos volumes que necessitam de proteção. Também por isso, muitas empresas têm investido em soluções alternativas, como snapshot e espelhamento de dados. Aproximadamente 65% das empresas entrevistadas em pesquisa conduzida pela consultoria Gartner tendem a aumentar o uso desses artifícios nos próximos anos.

Ocorre que a grande maioria das empresas questionadas pretende continuar utilizando sistemas tradicionais de backup. Até 2015, apenas 10% das grandes empresas deixaram de utilizar ferramentas de backup e restore convencionais em prol de soluções alternativas (snapshot e espelhamento). Segundo o mesmo relatório, dificilmente as ferramentas de backup (da maneira que conhecemos) deverão desaparecer, e aplicações específicas (hypervisors, bancos etc.) desempenharão um papel mais ativo e aprimorado nos backups,

com uma natural integração ao serviço de cópias de segurança.

Outro ponto que estimula o aprendizado e indica a continuidade do Bacula consiste na informação de que em 2014 ao menos 30% das organizações trocaram o fornecedor do software de backup por conta de frustrações relativas a custo, complexidade ou capacidade.

1.3 Mitos do backup²

1.3.1 O retorno do investimento (ROI) do serviço de backup

Em se tratando de um serviço relativo à área da segurança em tecnologia da informação, o backup não poderia faltar à regra: trata-se de um aspecto cujo retorno de investimento se mostra praticamente impossível de ser quantificado.

*Os benefícios são muito teóricos e incertos para que as empresas possam levá-los a sério como justificativa de investimento por si só. Para a maioria dos projetos em segurança da informação, é simplesmente impossível quantificar antecipadamente um ROI financeiro. Portanto, as organizações devem reagir com ceticismo a cálculos ou modelos de ROI divulgados por vendedores de soluções em segurança que não sejam substanciados por uma rigorosa pesquisa anterior, personalizada de acordo com o contexto de cada organização.*³

Ao longo do tempo, algumas companhias que experimentam um número limitado de falhas tendem a reduzir a atenção e o orçamento para backup, considerando eles, de certa forma, menos necessários. Ou, ainda, simplesmente deixam de fazer as devidas expansões indicadas pela gerência de capacidade, fazendo com que o sistema não atenda adequadamente à demanda.

Necessário ter em mente que backups e contratos de seguro não possuem muita diferença: “similar às apólices de seguro, são um investimento no qual é preferível deixar de precisar do que usá-los”, conforme as palavras do autor Preston de Guise.⁴

Portanto, a não ocorrência de demandas para a restauração de arquivos não pode, de forma alguma, constituir uma justificativa para que o serviço de cópias de segurança seja realizado de maneira precária – ou pior: que se deixe de fazê-lo.

1.3.2 Sistemas de tolerância a falhas não substituir backups

*Tolerância a falhas é a propriedade que permite que sistemas (em geral, computacionais) continuem a operar adequadamente mesmo após falhas em alguns de seus componentes. Se sua qualidade de operação diminui, a queda é proporcional à severidade da falha. A tolerância a falhas é propriedade inerente em sistemas de alta disponibilidade...*⁵

Dessa forma, o backup não vem, de forma alguma, constituir um elemento de tolerância a falhas. Entretanto, trata-se de um importante elemento da recuperação de falhas (inclusive o tempo necessário para a restauração de arquivos – e consequentemente de serviços – deve ser considerado uma métrica importante para a escolha do hardware e do software utilizados pela cópia de segurança).

Algumas empresas tendem a imaginar que sistemas tolerantes a falhas diminuem a necessidade por backups. Entretanto, os sistemas de tolerância a falhas são caros (nem

sempre podendo ser implementados em todos os pontos críticos do parque computacional) e em nenhuma hipótese podem ser considerados 100% eficazes.

“Nada corrompe mais rápido do que um espelho” – máxima que traduz o fato de que discos redundantes, por exemplo, não estão imunes à corrupção de dados, muito menos à deleção acidental de arquivos.

Portanto, os backups permanecem como uma necessidade imperativa para a grande maioria dos sistemas.

1.3.3 Existem sistemas de backup perfeitos

Os riscos da atividade de TI geralmente podem ser divididos em duas grandes áreas: os riscos operacionais e os comerciais. Em contrapartida, sabemos que nem todas as chances de perdas de dados podem ser evitadas – por mais caros que sejam os sistemas de proteção (ex.: um maremoto de proporções globais, uma guerra nuclear mundial etc.).

Deixando de lado as hipóteses catastróficas (nas quais provavelmente a restauração dos dados não será a coisa mais importante), deve ser feito um gerenciamento de capacidade constante no sentido de verificar a relação risco versus custo dos sistemas de backup, principalmente quando as opções são plurais. Exemplo: a armazenagem das cópias de segurança em edificação diversa de onde foram gerados, protegendo assim as mídias de um possível incêndio, seria preferível em relação à compra de um cofre antichamas? Ou ainda: seria interessante fazer um backup redundante (ou seja, duas cópias de segurança) de determinados dados?

Dito isso, é necessário levar em consideração uma série de fatores, como a criticidade da informação e o tipo, a maturidade dos usuários, a frequência de atualização de dados etc. E isso deve refletir na política de backup, nos acordos operacionais e de nível de serviço.

Backup é uma relação invertida de custo versus riscos.

1.3.4 Ferramentas de backup nativas dos SOs seriam suficientes ou mais confiáveis que as específicas?

Os softwares nativos dos sistemas operacionais carecem de uma qualidade essencial no âmbito dos backups – interoperabilidade. Dessa maneira, é impossível (ou excessivamente difícil) realizar cópias de segurança de vários servidores em apenas um storage – o que representa uma enorme economia com recursos de armazenamento. Ainda teríamos um maior custo pela manutenção e administração descentralizada de aplicações heterogêneas de backup.

Mesmo as ferramentas proprietárias de cópias de segurança são consideradas mais eficientes quando comparadas com as nativas – e, em relação à confiabilidade, não estariam há anos no mercado se não possuíssem essas qualidades. Diversas funcionalidades operacionais, de monitoração de backup e escalabilidade podem ser citadas:

- Possibilidade de backups e restores cruzados (ou seja, através da rede – de uma estação para outra diversa).
- Restores rápidos utilizando apenas as mídias necessárias para o trabalho.

- Catálogo para indexar os arquivos gravados, facilitando a busca por eles.
- Manipulação de robôs de fitas.
- Ferramentas de recuperação de desastres.
- Envio de alertas para sistemas de monitoração.

1.3.5 Scripts bem elaborados para backup podem suprir a necessidade de um software específico?

A elaboração de scripts para backup pode ser considerada um “retrabalho”, pois já existem aplicações bastante maduras desenvolvidas sob licenças livres. Ademais, tratar-se-ia de uma solução única, apenas utilizada no ambiente no qual se encontra em produção e impossível qualquer forma padronizada de suporte externo ou compartilhamento de experiências em outros cenários.

A continuidade da solução também estaria prejudicada se não houvesse uma detalhada documentação interna sobre os scripts, bem como os custos para o aprendizado seriam mais elevados do que com uma ferramenta de backup comum.

De igual sorte, as aplicações proprietárias de backup eventualmente realizam testes de performance que dificilmente seriam superados por um script. Segue o exemplo:

Em 2004, um vendedor de backup comercial publicou seu recorde de velocidade, realizando o backup de 10 TB das informações de um cliente real com uma taxa de transferência em nível de arquivo de 2,6 GB/s. É razoavelmente seguro dizer que poucos, para não dizer nenhum, administradores de sistemas desenvolveriam um script que atingisse uma performance de backup nessa escala.⁶

No entanto, scripts podem ser utilizados para lidar com softwares específicos que requerem procedimentos específicos de backup (a exemplo de dumps, parada da aplicação etc.), a fim de ter seus dados corretamente copiados. Uma vez que eles são geralmente chamados pelo software de backup, é possível monitorar sua execução e saída de erro de maneira integrada aos jobs de backup, tornando-se um processo seguro.

1.4 Glossário

- Retenção:** período pelo qual determinada informação não pode ser apagada do banco de dados ou da mídia de backup – a não ser por especial intervenção humana. A retenção começa a ser contada quando o job de backup termina, mas se o mesmo volume contiver outros jobs de backup, este só é reciclado quando os tempos de retenção de todos os jobs tiverem sido cumpridos.
- Expiração:** término do prazo de retenção, permitindo que os dados do job/volume sejam sobrescritos caso haja necessidade provocada por novos jobs de backup, por padrão e preferencialmente de maneira automática.
- Job:** trata-se da unidade de operação do backup. Eles podem ser jobs de **backup** – quando uma lista de pastas e arquivos é copiada para o storage de backup – ou jobs de **restore** – quando dados são lidos do storage e gravados novamente em alguma das máquinas com cliente do software de backup instalado. Existem outros subtipos de jobs: **cópia e migração**, que faz uma segunda cópia ou move jobs de backup já

terminados para outro dispositivo de armazenamento; e **verificação**, utilizado para verificar a consistência e integridade de diversos elementos do sistema de backup: storage, catalog, client files etc.

• **Purge**: ato de eliminar os dados do catálogo a respeito de determinado volume (files, jobs etc.) do banco de dados do Bacula. O purge não afeta imediatamente os dados gravados na fita, mas o volume ganha o status “purged” e, assim que haja demanda de mais espaço para a gravação de novos jobs de backup, ele pode ser automaticamente reciclado, ou seja: ter seus dados descartados e sobrescritos. Se o desejado é forçar que o Bacula reutilize um volume já gravado ainda que esteja dentro do tempo de retenção, **purge** é também o nome do comando para forçar que ele seja reciclado.

• **Prune (podar)**: além da retenção do volume (dados de backup), há também outras retenções que afetam as informações de banco de dados do software de backup. No caso do Bacula, retenções dos índices de arquivos e trabalhos de backup (diretivas **File** e **Job Retention** no recurso Client do bacula-dir.conf⁷). Essa configuração é utilizada para limitar o crescimento do banco, descartando esses índices ao custo de menos opções para a busca de arquivos e granularidade da restauração. Exemplo: se a informação de arquivo é “podada”, seria impossível restaurar um único arquivo de todo o backup, a menos que seja utilizado um procedimento mais moroso e manual de recuperação de desastres do Bacula (como bextract ou bscan). Essa ação por padrão é automática (AutoPrune = yes, recurso Pool, bacula-dir.conf), ou manual, quando o administrador precisa liberar algum espaço (comando **prune** no bconsole para liberar espaço quando desejado).

• **Volume**: trata-se da divisão lógica (espécie de caixa) reservada para o armazenamento de dados de backup. Mesmo volumes gravados em **disco rígido** são arquivos em sequência de bytes, escritos e lidos sequencialmente. Seu tamanho cresce dinamicamente quando novos jobs são escritos neles, até um tamanho de volume máximo definido. Definir um tamanho máximo é importante para que haja granularidade na reciclagem dos volumes antigos em um mesmo disco após a retenção ter expirado, uma vez que a reciclagem de um volume é uma ação atômica (tudo ou nada). Diferentemente dos discos que podem e devem ter vários volumes de backup, **fitas magnéticas** podem ter apenas um volume por fita, cujo tamanho coincide com sua capacidade máxima. Não há motivo, portanto, para limitar o seu crescimento.

• **Pool**: conjunto de volumes com características semelhantes (ex.: tempo de retenção). De igual sorte, trata-se de um dispositivo de segurança (se o backup utilizando determinada pool é agendado, não poderão ser utilizados volumes de outra pool). O volume herda as características da pool. Um erro muito comum é pensar que você é capaz de determinar o volume específico no qual uma tarefa de backup será gravada, mas isso seria muito manual. As tarefas de backup são sempre submetidas a uma **pool**, e o algoritmo do software de backup é quem escolhe o melhor volume para armazenar os backups em execução.⁸

• **FileSet**: lista de diretórios (ou arquivos) para o backup. No Bacula eles são objetos, de maneira que podem ser reutilizados para o backup de máquinas diferentes (se a lista de pastas e arquivos a ser backupeada é exatamente a mesma) ou pode simplesmente ser configurado um FileSet diferente para cada máquina ou job de backup. Normalmente existem duas sublistas dentro do FileSet: uma para incluir diretórios e arquivos, outra

para exclusão. Diretórios **incluídos** são copiados de forma recursiva e, por isso, é possível também **excluir** qualquer subcaminho indesejado ou arquivo a partir da seleção incluída (por exemplo: /proc, /tmp, se o backup da raiz do Linux "/" tiver sido o caminho incluído). No Bacula também possível combinar nomes e caminhos de arquivos com curingas e expressões regulares, usando diretivas específicas no sub-recurso **Options**, recurso **FileSet** (bacula-dir.conf).

•**Catálogo:** é o banco de dados responsável por armazenar um índice de todos os arquivos armazenados pela ferramenta de backup, permitindo visualizar, navegar e selecionar arquivos dos jobs de backup, ainda que os dados estejam em fitas não inseridas nos drives. Uma vantagem do Bacula é que ele suporta três bancos diferentes (e livres) para armazenamento do catálogo, tendo sido o primeiro software a utilizar padrão SQL para seu catálogo (**PostgreSQL** e **MySQL**, sendo que SQLite teve o suporte descontinuado). Essas características proporcionam uma gama de recursos para recuperação de desastres do próprio serviço de bancos de dados e a possibilidade de mineração destes (machine learning), ao contrário de outras soluções proprietárias, que geralmente contêm uma solução obscura fechada de banco integrada.

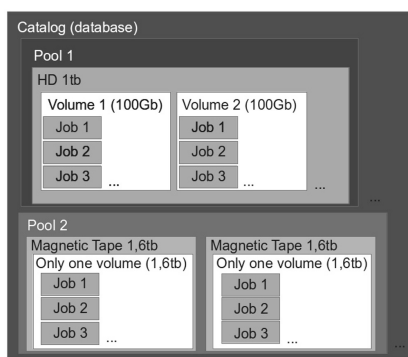


Figura 1.1. Representação gráfica dos principais objetos do Bacula

A **Figura 1.1** representa a hierarquia dos elementos no banco de dados do Bacula. Os jobs estão contidos nos volumes, que podem ser uma fita (no caso, sempre um volume por fita) ou podem estar contidos em disco rígido. Cada job de backup pode estar em apenas um volume (monovolume) ou em vários (multivolume). Todos os volumes precisam estar contidos em um conjunto (pool). Normalmente, volumes de diferentes dispositivos de armazenamento (storages) estão em diferentes pools. Para um mesmo storage, no entanto, você poderá ter várias pools (conjuntos de volumes), com a finalidade usual de ter diferentes tempos de retenção para conjuntos de volumes diferentes (backup escalonado/estratégia GFS). Por fim, informações de storage/pools estão todas contidas dentro de um mesmo catálogo. Eventualmente, no caso de gargalos de performance no banco de dados, você pode ter um segundo catálogo, desde que contenha informações de storage/pools/volumes/jobs/etc. diferentes.

•**GFS (Grandfather, Father, Son⁹):** estratégia de rotação de fitas mais utilizada pelos administradores no mundo. Os backups de um nível menor podem ser reciclados desde que seja feito um backup com nível maior, ou seja: que tenha um tempo de retenção consideravelmente maior. Dessa maneira, sacrifica-se granularidade de restauração

(opções de modificações de arquivos, por exemplo, de dias) em prol da economia de espaço no storage de backup. Tradicionalmente, divide-se em backups diários, semanais e mensais respectivamente, que são implantados através da configuração de pools distintas (ex.: com esses mesmos nomes). Esse assunto será detalhado nos próximos capítulos.

•**Janela de backup:** consiste no tempo máximo de duração que pode ser executado um job de backup, de maneira a não gerar impactos para o respectivo serviço, dono dos dados, restrições de negócio ou evitar problemas no uso concorrente das aplicações backupeadas. Por isso, na maioria das vezes, os backups são agendados logo após o expediente administrativo, momento em que os dados já não devem sofrer tantas alterações, aumentando, assim, a fidelidade da cópia de segurança. Lógico que determinadas aplicações possuem uma grande atividade ininterrupta (24x7) – nestas, nas quais não há janela de backup, o horário de realização do backup torna-se menos importante.

•**Política de backup:** trata-se de um documento corporativo que descreve os requisitos dos processos de backup, riscos existentes e ações para mitigá-los, além de outros procedimentos de recuperação de desastres. Deve também ter prazo de validade, para forçar o controle e a reavaliação dos riscos, e será detalhada nos próximos capítulos.

•**MTBF (Mean Time Between Failure)**¹⁰: esta é uma especificação de hardware importante, mas também complexa (os fabricantes diferem na maneira do cálculo dessa métrica). Ela se refere ao “tempo previsto decorrido entre falhas inerentes de um equipamento durante a sua operação”¹¹. Em outras palavras: é a soma dos períodos em operação dividida pelo número de falhas observadas. **MTBF é diferente de vida útil esperada!** Ex.: um SSD aleatório com 2.000.000 horas de MTBF costuma considerar um uso diário de 8 horas para 1.000 equipamentos SSD. **2.000.000/8 horas por dia = 250.000/1.000 unidades** = uma falha esperada cada 250 dias.¹²

1.5 Níveis de backup

Existem três principais níveis de backup: **completo, diferencial e incremental**. Os trabalhos de cópia e de migração são considerados subníveis para este estudo, uma vez que apenas repetem ou movem backups originais, mas na lógica Bacula são considerados jobs de backup independentes (apenas para referência futura).

Além da definição por escrito que será dada mais à frente, uma boa maneira de entender a geralmente negligenciada diferença entre backups **diferenciais** e **incrementais** é graficamente (**Figura 1.2**).



Figura 1.2. Backup incremental x diferencial

Segue a descrição dos principais **níveis de backup**:

•**Full:** sempre fará a cópia de todos os diretórios e arquivos definidos para inclusão no FileSet, independentemente de qualquer outra condição. Será sempre o primeiro backup de qualquer servidor, ainda que um incremental ou diferencial for submetido (pois não há backups passados para comparar). Quando isso acontece, o software de backup automaticamente converte o job para full, e o mesmo ocorre (por padrão) quando qualquer modificação ocorre no FileSet (o administrador inclui ou exclui pastas a serem backupadas, por exemplo). Como é um backup mais longo, geralmente é agendado para começar na sexta-feira ou no sábado, tendo toda a janela do final de semana para ser realizado.

•**Diferencial:** faz o backup apenas dos arquivos modificados a partir do último backup full terminado, por isso normalmente tem um tamanho consideravelmente menor. Exemplo: um backup full ocorreu no sábado; um backup diferencial realizado na segunda só conterá os dados alterados ou criados na segunda; se na terça-feira for gravado outro backup diferencial, ele novamente vai gravar os arquivos alterados ou criados desde que se gravou o backup full, isto é, os arquivos de segunda e terça. Dumps de bancos de dados e exportações de discos de máquinas virtuais sempre serão copiados no diferencial se gerados depois do último full, o que pode equiparar o tamanho do diferencial a um backup completo (a não ser que tenha um plug-in especial para backup da aplicação diferencial em nível de blocos¹³). A **regra de ouro** para restauração utilizando backup diferencial será a leitura do **último backup full e do último diferencial**, permitindo restaurar todos os arquivos de um servidor em um determinado ponto do tempo. Observe que a **preservação do full** é essencial, pois existem arquivos que não foram modificados e que só estão presentes nele. Também por isso não é recomendado fazer apenas um backup full e somente diferenciais depois, pois a eventual perda do full irá comprometer a restauração de todos os outros backups diferenciais.

•**Incremental:** este faz backup de ainda menos arquivos do diferencial ao longo do dia, uma vez que apenas copia os arquivos que foram alterados desde o último backup, independentemente do seu nível. Em outras palavras, os backups incrementais copiam estritamente os arquivos que não estão presentes em qualquer backup anterior. A fim de restaurar a partir de todos os arquivos de clientes em um determinado momento, o software de backup terá que ler não só o último completo, mas todos os outros backups incrementais posteriores. Dessa forma, a perda de apenas um backup incremental teria grande impacto, uma vez que este pode conter a única versão de um arquivo necessário para uma restauração de uma máquina inteira. O backup incremental também pode requerer uma quantidade numerosa de diferentes volumes de leitura na restauração. Isso não traz problemas quando o backup é armazenado em disco, mas para fitas é bastante prejudicial ao tempo de restore. Mesmo que haja um robô de fitas, poderiam ser necessários vários posicionamentos e trocas das fitas em pontos específicos, causando também mais desgaste ao equipamento. Em se falando de drives manuais, então, seria o pior cenário. Algo importante a considerar ao escolher entre o backup diferencial ou o incremental é que, quando o tempo passa, os arquivos alterados tendem a ser basicamente os mesmos. Muito provavelmente eles são os mesmos arquivos de log, dumps de banco de dados, algumas folhas de cálculo a partir do servidor de arquivos etc. (a menos, claro, que estejamos lidando com uma carga de

trabalho muito incomum). Isso posto: a economia de armazenamento ao usar backups incrementais pode não compensar a complexidade e o impacto de eventual perda de dados do backup.

•**Cópia:** este é o backup do backup. Ele basicamente gera outra cópia de um job terminado, apenas lendo as informações do storage de backup. Os trabalhos de cópia são muito úteis para permitir backup off-site, disk2disk2tape, mas há uma série de outros possíveis usos (por exemplo: de backup cópia de disco mais rápido para um dispositivo de armazenamento de backup mais lento). Se o Bacula eventualmente perde informações de seu catálogo sobre os jobs de backup originais, ou se eles foram simplesmente expirados e reciclados, o job de cópia é automaticamente selecionado para restauração de dados caso requisitado.

•**Migração:** é o mesmo que a cópia, mas descarta o job original depois que o trabalho de migração é concluído com êxito. Um exemplo de uso seria mover jobs de backup a partir de um hardware de armazenamento obsoleto para um novo, ainda que sejam fitas com capacidades diferentes (por exemplo: uma biblioteca de fitas com capacidade de 800 GB poderia conter informações de quatro fitas de 200 GB mais antigas).

1.6 Mídias de backup

1.6.1 Fitas magnéticas

As fitas magnéticas são confiáveis, relativamente rápidas, economizam processamento (já que realizam a **compressão dos dados via hardware**) e podem ser consideradas a melhor mídia para **armazenamento de longo prazo** (30 anos de MTBF). A leitura e a gravação são sequenciais e, portanto, adequadas aos procedimentos de backup que normalmente já salvam os dados em sequências de bytes.

As capacidades das fitas magnéticas normalmente são expressas em dois valores diferentes (ex.: **400/800 GB**). O **primeiro valor** diz respeito à capacidade máxima (nativa) se a compressão via hardware não estiver habilitada. O **segundo valor**, caso esteja habilitada. **Se os dados copiados já forem compactados** (ex.: pelo software de backup ou em um formato naturalmente compactado, como mp3, mp4, gz, zip etc.), a capacidade real da fita vai estar perto da capacidade nativa (ex.: 400 GB), pois a compressão via hardware terá basicamente nenhum ganho.

A escrita e a leitura dos drives de fita atuais não são necessariamente mais lentas do que os discos rígidos: **em muitos cenários elas são ainda mais rápidas!** No entanto, algumas restaurações individuais de arquivos podem ser mais lentas devido à necessidade de carregamento de fita no drive e ao posicionamento¹⁴, a fim de ler apenas um ou alguns trechos da mídia. Atualmente as fitas **LTO (Linear Tape-Open)** possuem as seguintes capacidades especificadas:

- 400/800 GB (LTO-3); 80 MB/s gravação.**¹⁵
- 800 GB/1,6 TB (LTO-4), 120 MB/s gravação.**
- 1,5/3 TB (LTO-5), 140 MB/s gravação.**
- 2,5/6,25 TB¹⁶ (LTO-6), 160 MB/s gravação.**
- 6,0/15 TB (LTO-7), 300 MB/s gravação.**

Tanto o **drive de fitas manuais** quanto os **robôs de fitas** podem ler e escrever fitas, às vezes até de gerações anteriores (ex.: unidades LTO4 podem ler também LTO2 e 3). No entanto, as unidades manuais devem ser evitadas a todo custo: elas sempre vão exigir intervenção humana (mais cara a cada dia) e vão deixar os backups e as rotinas de restauração invariavelmente mais lentos por conta da exigência da troca manual de mídias.

Robôs de fita têm, em geral, um custo inicial mais elevado, mas trazem a possibilidade de aumento de capacidade virtualmente infinito e por um custo menor através da compra de mais fitas magnéticas, ainda que fiquem armazenadas fora do robô. Elas podem ser inseridas e removidas de acordo com a demanda por novos backups e trabalhos de restore.

1.6.2 HDD

Nos últimos vinte anos, unidades de disco rígido melhoraram bastante no quesito resiliência dos dados e houve grande redução do custo por gigabyte. Essas mudanças impulsionaram sua utilização como armazenamento de backup em várias empresas ou mesmo soluções híbridas (por exemplo, backups **disk-to-disk-to-tape**¹⁷).

A solução de backup baseada em disco tem menor investimento inicial porque é possível comprar apenas alguns discos/espço em storage para início das rotinas de backup e acrescentar mais quando houver crescimento da demanda.

Os discos rígidos têm a vantagem de estar sempre montados para o sistema operacional, de modo que o software de backup pode se mover entre diferentes discos ou volumes de backup com praticamente nenhuma latência, em contraste com as bibliotecas de fitas.

O acesso randômico torna mais rápido ler qualquer posição dos volumes de backup, mesmo que ele tenha sido escrito como uma sequência de bytes, tornando a restauração de pequenos arquivos significativamente mais rápida do que na fita.

Sobre velocidades de gravação de backup, elas diferem de acordo com os modelos de discos rígidos e devem ser analisadas a cada caso. No entanto, é possível estimar que a maioria dos exemplares modernos podem atingir o seguinte valor¹⁸:

- 80-200 MB/s, capacidade de escrita.¹⁹

Network Attached Storages (NAS) são conjuntos de discos flexíveis e escaláveis que podem ser facilmente particionados e disponibilizados para o **Bacula Storage Daemon (serviço de armazenamento do Bacula)**, através do protocolo rápido **iSCSI (deve-se evitar NFS ou CIFS)**.

1.6.3 SSD

Apesar dos custos e das dúvidas iniciais em termos de integridade de armazenamento de dados a longo prazo, os SSDs atuais são mais rápidos que a maioria das fitas LTO da geração atual e que os discos rígidos tradicionais. Dependendo do modelo, o MTBF anunciado pode ser de até 2 milhões de horas! Isto é, na verdade, mais do que o especificado para alguns HDDs. O SSD pode ser considerado uma mídia factível de backup em poucos anos.

Hoje em dia, os discos sólidos podem ser usados como área de spooling (ou cache) do backup, possibilitando a cópia mais rápida de dados a partir de clientes com uma janela de

backup mais restrita, ainda que os dados do spooling precisem ser copiados para uma fita mais lenta ou HDD (em ato contínuo).

A velocidade do SSD também depende do modelo.

- 200-550 MB/s, capacidade de escrita.²⁰

1.7 Local de armazenamento do backup

• **Backup on-site:** se os storages ou mídias de backup são armazenados no mesmo edifício ou muito perto das máquinas que contêm os dados originais, este backup deve ser considerado on-site. Este método pode ser prático, porém é mais suscetível a perigos físicos que afetem o local (incêndios, inundações, furto de equipamentos etc.). Neste caso, cofres à prova de fogo poderiam ser usados para mitigar alguns dos riscos no local.

• **Backup off-site:** Backups off-site estão no cenário ideal de acordo com a **ISO 27002**. Na verdade, eles recomendam **15 km** de distância do original para dados de backup (o que, aliás, é a zona de segurança de um impacto de uma bomba de **1 megaton**).

Hoje em dia os serviços de nuvem podem ser uma solução para algumas pequenas cargas de trabalho de backup, uma vez que as taxas de upload das conexões de internet são geralmente baixas e não suportam uma transferência de grandes quantidades de dados.

¹ Dados do relatório Gartner “The Future of Backup May Not Be Backup”.

² A ideia de discorrer sobre mitos do backup foi retirada do livro de Preston de Guise, “Enterprise Systems Backup and Recovery: a corporate insurance policy”.

³ Rodrigo Faustini.

⁴ GUISE, P. de. **Enterprise Systems Backups and Recovery:** a corporate insurance policy, p. 7.

⁵ <[http://pt.wikipedia.org/wiki/Tolerância_a_falhas_\(hardware\)](http://pt.wikipedia.org/wiki/Tolerância_a_falhas_(hardware))>.

⁶ GUISE, P. de. **Enterprise Systems Backup & Recovery:** a corporate insurance policy, p. 15.

⁷ File e Job são também os nomes das tabelas do banco de dados do Bacula que contêm essas informações.

⁸ Seria possível forçar e escrever em um volume específico criando uma pool para cada volume, mas a administração provavelmente seria bem complexa e não muito recomendável.

⁹ Avô, pai e filho. Isso é uma referência ao uso de diferentes níveis de retenção (backup escalonado).

¹⁰ Também representada pela letra grega θ .

¹¹ JONES, J. V. **Integrated Logistics Support Handbook**, p. 4.

¹² Disponível em: <<http://superuser.com/questions/950825/ssd-2-million-hour-mtbf-how-is-this-proven>>.

¹³ Neste caso, apenas no Bacula Enterprise: <<http://www.bacula.com.br>>.

¹⁴ É possível personalizar o tamanho máximo de pedaços de fita que o Bacula precisará ler para restaurar arquivos. Ele pode posicionar a fita em múltiplos desse valor e começar a ler dados a partir desse ponto (diretiva Maximum File Size, recurso Device, bacula-sd.conf).

¹⁵ Valores para dados gravados em fita não compactada. Disponível em:

<http://www.theregister.co.uk/2015/09/16/ito_has_15tb_gen_7_tape_format/>.

¹⁶ De acordo com a compressão anunciada de 2,5: 1, em vez de proporção 2:1 de gerações anteriores.

¹⁷ O primeiro backup é para disco e, em outra agenda, copiada para a fita.

¹⁸ É possível testar a capacidade de gravação de um disco no Linux, por exemplo:

```
dd if=/dev/zero of=/tmp/output bs=8k count=10k; rm -f /tmp/output
```

¹⁹ Disponível em: <<http://www.tomshardware.com/charts/hdd-charts-2013/-05-Write-Throughput-Maximum-h2benchw-3.16,2903.html>>.

²⁰ Disponível em: <<http://ssd.userbenchmark.com>>.

2

Passos para Escolha e Implantação de um Sistema de Backup

Datacenters corporativos se tornam mais complexos a cada dia e com um crescimento na quantidade de dados imprevisível. No entanto, os procedimentos a seguir objetivam ser genéricos o suficiente para ajudar a encontrar um processo mínimo de backup e restauração que possa ser aprimorado e melhorado ao longo do tempo.

2.1 Existe uma política de backup?

A política de backup é um documento que fornece os parâmetros mínimos aos administradores de sistemas para que possam planejar e implantar um sistema de backup. Sem uma política é impossível determinar se um ambiente de backup é confiável, suficiente ou mesmo adequado.

Caso não haja uma política de backup corporativa, esse é o momento para construir uma: existe inclusive um modelo com mais detalhes no próximo capítulo.

2.2 Quais são os serviços e negócios backupeados?

Quais são as necessidades de backup e restauração do seu negócio ou dos serviços de TI? Existem acordos operacionais ou de nível de serviço estabelecidos para o backup? Como esses acordos afetam o processo de backup e restauração? A política de backup reflete esses requisitos?

Características específicas de negócios podem exigir diferentes necessidades de backup. Por exemplo: backups diários podem não ser suficientes para um dado que é crítico em uma organização e talvez requeira backups de hora em hora.

2.3 Quais aplicações específicas e sistemas operacionais precisam de backup?

Cada aplicação tem diferentes maneiras e necessidades de fazer uma cópia de segurança utilizável. Na maioria das vezes, existem muitas opções para a mesma aplicação. O administrador de backup deve saber quais são as opções, testar e decidir o que é melhor para uma eventual e adequada restauração.

Exemplos desse processo podem ser: dump do banco de dados; backup do arquivo de transações on-line de banco de dados; parar o serviço de banco de dados a fim de fazer a

cópia de seus arquivos (datafiles); etc. Note também que os procedimentos podem variar de acordo com os diferentes desenvolvedores (MySQL, PostgreSQL, Oracle etc.). E estamos falando apenas de bancos de dados.

Imagens de máquinas virtuais são ainda mais complexas em termos de backup. Geralmente, requerem a execução de um snapshot para cada VM, exportar para um diretório qualquer e, em seguida, copiá-los com o software de backup.

A maioria desses procedimentos de backup é feita através de scripts **Shell**, **Perl**, **Windows** etc. e consomem muito espaço em disco temporário. No entanto, existem opções melhores: a maior parte dos softwares de backup de nível empresarial (por exemplo, Bacula Enterprise) fornece plug-ins prontos e amigáveis que automatizam o backup e a restauração dessas aplicações, suprimindo a necessidade de utilização desses scripts. Alguns plug-ins também fornecem recursos mais avançados, tais como backups diferenciais das mencionadas aplicações em nível de blocos de bytes etc.

O **Bacula Community** prevê também uma outra opção: um plug-in genérico (**bpipe**) que pode receber fluxos de dados de uma aplicação Linux automaticamente para um pipe (FIFO), para o serviço de armazenamento de backup. Isso pode ser utilizado para os mencionados dumps de banco de dados e exportações de máquinas virtuais.²¹

Da mesma forma, os sistemas operacionais devem ter clientes que são suportados pelo software de backup – e o Windows (em especial) também possui procedimentos específicos para registro e outras informações importantes de recuperação de desastres. Verifique se o seu software e suas habilidades de backup abrangem todos os quesitos.

2.4 Qual o tamanho do backup?

Esta questão aparentemente simples pode não ser prontamente respondida pela maioria das empresas. É preciso considerar não apenas um único tamanho de backup completo (normalmente comprimido), mas também o diferencial (ou os incrementais), lembrando que eles ocuparão o armazenamento durante todo o período das retenções configuradas (na estratégia de GFS, a soma dos backups diários, semanais e mensais). Isso pode resultar em requisitos de armazenamento grandes, dos quais nem sempre as pessoas que elaboram as políticas de backup estão conscientes.

Além disso, é importante considerar que os projetos planejados vão exigir ainda espaço em armazenamento adicional por conta do crescimento natural no uso dos sistemas corporativos previamente instalados (algum número mágico entre **20-25%** por ano, podendo sempre variar de acordo com as aplicações do caso concreto).

2.5 Qual é a capacidade de armazenamento atual?

Não só o espaço de armazenamento é importante para backups, como também a velocidade de escrita. Imperativo saber se o seu disco ou fita de armazenamento tem performance suficiente para realizar os backups dentro da janela de backup. Normalmente, ter mais de um storage de discos ou unidade de fita aumenta a capacidade geral de escrita, desde que você consiga ter backups simultâneos e recursos suficientes. Lembre-se de que, em qualquer

hipótese, gargalos podem aparecer: capacidade de rede, velocidade de leitura de arquivos no disco da máquina do cliente etc.

As bibliotecas de fitas possuem grande capacidade de armazenamento ao permitir ler e gravar uma vasta quantidade de fitas, mesmo que não acessíveis de maneira imediata (fora do robô de fitas). Se os drives presentes fornecem a capacidade de escrita necessária e o hardware dentro do tempo de MTBF, entende-se que o equipamento é adequado à demanda. Note também que as unidades de fita normalmente comprimem os dados por padrão – e que é mais eficiente que a compressão via software.

Armazenamentos de disco são muito flexíveis, sendo relativamente simples reservar uma área em storage fixa e torná-la disponível para o daemon de armazenamento do software de backup através de um ponto de montagem. O tipo de sistema de arquivos pode afetar escrevendo capacidade de leitura e gravação (hoje XFS é recomendado) e a compressão, no caso, precisa ser feita via software para obter economia. Um dilema que afeta o espaço de armazenamento em disco seria o uso de níveis de proteção RAID maiores que 0: alguns analistas acham que, por se tratar de dados de backup, não haveria necessidade de uma proteção adicional com redundância dos dados nos discos de backup; outros pensam que esses custos são compensados com a maior tolerância a falhas.

2.6 O software de backup tem compatibilidade com o armazenamento?

Nós já sabemos que o hardware de backup é essencial para alcançar um bom desempenho, mas uma pergunta que muitos novos administradores têm é sobre o software de backup e sua compatibilidade com determinado hardware.

Os storages de backup, incluindo bibliotecas de fitas, são em geral equipamentos padronizados e com grandes chances de trabalhar com qualquer software de backup. Na verdade, é mais provável encontrar incompatibilidade entre o equipamento e a placa controladora SCSI, ou mesmo de armazenamento e sistema operacional. Há um consórcio de fabricantes de hardware que normalmente define especificações de bibliotecas de fitas, e o atual é o LTO.

Isso posto, apenas algumas mudanças de configuração são necessárias para a configuração desses dispositivos, mais comumente para robôs de fitas de modelos mais antigos, como a configuração de “sleep” entre os comandos de carga e descarga das fitas nos drives.

2.7 Qual formato de armazenamento utilizar para o backup?

A maioria dos softwares de backup armazena os dados na forma de sequências de bytes (chamados volumes), ainda que sejam gravados em discos rígidos. Isso privilegia velocidade de gravação ao custo de tornar mais custosa (lenta) a restauração de um ou alguns arquivos de um backup.

Mas a questão principal, no entanto, trata de saber se o software de backup utiliza um **formato proprietário** de gravação ou não²². Softwares com formato fechado de gravação geram uma enorme quantidade de dados que só podem ser restaurados com o mesmo

software (**aprisionamento tecnológico**), ficando muito caro migrar esses dados para outra solução depois que centenas ou milhares de jobs de backups já foram gravados.

Vários fornecedores tiram proveito desse aprisionamento ao negociar a renovação de licenças de software, chegando inclusive a conceder o primeiro ano “gratuitamente”, sabendo que após esse período será mais difícil para o consumidor realizar qualquer migração.²³

2.8 Existe algum legado que precise ser tratado?

Normalmente, existem duas formas de lidar com o legado – backups executados através de outros softwares proprietários:

- **Manter uma instalação mínima do software anterior e acesso ao storage** para eventual restauração de dados, até que os backups realizados por aquele tenham sido expirados.
- **Restaurar alguns ou todos os backups realizados pelo software anterior** para uma máquina qualquer e daí realizar novo backup com o novo software. Alguns softwares (como o Bacula) permitem modificar a data e o horário da execução desse job “migrado²⁴” manualmente no banco de dados. Isso pode ser feito para preservar a data de realização do job original feito pelo software antigo.

2.9 O que mais considerar?

Os administradores devem ter em mente que os sistemas de backup podem ser de difícil migração e, às vezes, excessivamente caros na renovação de licenças que envolvam a tarifação pelo tamanho dos backups, por exemplo. Eles devem avaliar os custos e a eficiência do backup para vários anos (por exemplo, cinco), e não apenas a necessidade imediata.

Deve-se considerar ainda que as especificações dos equipamentos (inclusive MTBF) devem cumprir as retenções definidas na política de backup, que os riscos de negócio precisam ser mitigados, que o crescimento não planejado da demanda pode acontecer e como redimensionar rapidamente o sistema de backup se for o caso.

²¹ A maioria desses métodos é explicada em capítulos posteriores.

²² Ambos Bacula Community e Enterprise armazenam backup em formato aberto, que poderiam, no pior caso, ser lidos por outros softwares tradicionais do Unix/Linux.

²³ É uma cilada, Bino!

²⁴ Isso pode ser feito através da console do MySQL, PostgreSQL ou comando query do bconsole. Ex.: `update job set starttime='2015-01-01 00:00:00' where jobid='529';`

Política de Backup e Melhores Práticas

3.1 Política de backup

A política de backup é um documento que descreve os requisitos básicos de backup, riscos e alguns procedimentos, devendo ser genérica o suficiente para não aprisionar o corpo técnico da empresa a uma tecnologia específica (que sempre muda).

Ter uma política vigente é uma prática recomendada e deve ser considerada “lei” ao fazer backups em uma determinada corporação. É impossível planejar um sistema de backup sem ter uma política. Se não houver nenhuma informação declarada sobre tempos de retenção de backup (por exemplo), seria impossível para o administrador do sistema saber o tamanho necessário dos storages, robôs de fitas ou mesmo a quantidade de fitas para armazenar backups para um determinado período de tempo. O crescimento vegetativo também seria imprevisível.

A falta de uma política pode causar também uma situação de insegurança jurídica. Por exemplo: os usuários sabem sobre o período de tempo que podem confiar em uma restauração de dados perdidos? Os backups são suficientes para cumprir os contratos operacionais e de nível de serviço?

A política deve trazer também mapeados os riscos de segurança óbvios (e não tão óbvios) que podem afetar o ambiente de backup. Eles são geralmente identificados, classificados e têm ações de mitigação previstas em reuniões de gerenciamento de riscos simplificado. Incluir os riscos na política de backup garante que gerentes, executivos de nível tático e até estratégicos também sejam responsáveis por eles. Na verdade, a política de backup geralmente é assinada por um membro da alta liderança e consiste em uma ótima maneira de justificar ou buscar investimentos adequados para o backup.

Dito isso, essas são as quatro (ou cinco) informações básicas que uma política de backup deve conter (embora não seja uma lista exaustiva):

- Matriz de responsabilidades: atribuição de atividades de backup e restauração ao time operacional.
- Periodicidade do backup e tempos de retenção, divididos por cada tipo de informação digital ou aplicação.
- Identificação dos riscos, classificação e ações para sua mitigação (gestão simplificada de riscos).
- Data de validade da política de backup: riscos e requerimentos do backup devem ser reavaliados após um período (por exemplo: 1 ano).

• Processo de descarte de mídias obsoletas (incineração, picotamento, ofuscação de dados residuais por software etc.) – a não ser que já esteja contemplado dentro de uma política de segurança geral.²⁵

3.1.1 Um modelo de política de backup

Política de Backup e Restauração Corporativa

Versão: x.x

1. Introdução

Este documento estabelece políticas de cópia de segurança (backups), armazenamento e restauração de dados digitais presentes nos servidores e estações de trabalho dessa organização.

Os proprietários dos dados devem ter ciência dos termos deste documento e dos tempos de retenção a serem praticados para cada tipo de informação, e o corpo técnico deve assegurar as ações necessárias para que estes sejam cumpridos.

Este documento deve ser divulgado aos usuários, gestores e alta direção.

2. Matriz de Responsabilidades

I. O administrador de backup é responsável por conhecer a política de backup, identificar acordos e instruções normativas que se relacionem com esta, planejar e executar operações de backup para todos os computadores da organização, preferencialmente depois do horário administrativo. Ele também é responsável pela monitoração, correta execução dos backups, ajustes, reexecução dos backups em caso de falha e aderência a todos os requisitos operacionais requeridos, tais como janela de execução, procedimentos específicos de backup de determinada aplicação, dentre outros.

II. O administrador de restauração é responsável por receber e processar demandas eventuais de restauração de dados, por realizar testes de restauração periódicos de cada tipo de aplicação, mídia e storage, ao menos a cada 6 (seis) meses. Ele pode gerir, de igual sorte, sistema restrito de autoatendimento para que os próprios donos de dados possam executar o trabalho de restauração dos arquivos necessários.

III. A restauração de arquivos deve poder ser executada em no máximo 4 (quatro) horas a partir da verificação de perda dos arquivos, salvo acordos operacionais ou de nível de serviço mais restritivos.

IV. Preferencialmente, as atividades de administração de backup e restauração serão executadas por pessoas diferentes.

V. Gestores de backup e restore devem executar avaliações de capacidade periodicamente, além de avaliar o impacto de projetos futuros na execução dos backups, prever o crescimento dos backups e indicar as melhorias de hardware e software necessárias ao cumprimento dessa política.

3. Periodicidade do Backup e Tempos de Retenção

I. Os seguintes dados e aplicações serão copiados de acordo com a Estratégia de Rotação GFS Clássica, em backups diários diferenciais, semanais, mensais e anuais (históricos) full, com uma retenção mínima de 7, 31, 365 dias e 5 anos, respectivamente:

Dados de servidores de arquivos.

Servidores de bancos de dados.

Aplicações web, logs e configurações.

Backup de máquinas virtuais em nível de imagem.

Configuração ordinária de servidores.

Logs.

Servidor de backup, configurações e banco de dados.

Mensagens e banco de dados do serviço de e-mail.

(etc.)

II. As fitas que contenham os backups anuais (históricos) serão retirados do robô de fitas e armazenadas em local mais seguro.

4. Gestão Simplificada de Riscos

I. Para a finalidade deste documento, a seguinte notação é utilizada para o registro de riscos levantados:

Risco #: [fator gerador do risco]

Impacto: [baixo, médio, alto]

Tipo do Risco (D, C, I): [Disponibilidade, Confidencialidade, Integridade]

Ação: [procedimentos para mitigação ou eventualmente aceitação do risco]

II. Durante o ciclo atual de gestão de riscos, estes foram os levantados e classificados para fins de mitigação:

Risco 01: falta de condições ideais de umidade, acesso, proteção contra incêndio e temperatura para armazenamento de fitas históricas fora do robô de fitas.

Impacto: médio

Tipo do Risco (D, C, I): Disponibilidade, Integridade

Ação: aquisição de um cofre de fitas.

Risco 02: crescimento exagerado da demanda de espaço em armazenamento para backup.

Impacto: médio

Tipo do Risco (D, C, I): Disponibilidade

Ação: comprar mais fitas.

Risco 03: falta de espaço temporário para geração de dump/mecanismo não automatizado de backup do PostgreSQL.

Impacto: médio

Tipo do Risco (D, C, I): Disponibilidade

Ação: comprar mais fitas.

5. Procedimento de Descarte de Mídias Obsoletas

Antes de descartadas, as mídias de backup serão perfuradas, incineradas ou processadas por software específico para ofuscação de dados residuais (ex.: Eraser), de maneira a impossibilitar a recuperação de dados residuais por terceiros.

6. Validade desta Política

Esta política será válida até um ano após a sua assinatura. Antes de expiradas, os riscos e ações devem ser reavaliados para a publicação de uma nova versão.

3.2 Melhores práticas do backup

Alguns procedimentos recomendados para a produção de backups confiáveis:

- Testar tudo!
 - Testar a configuração e conexão com novos clientes de backup após “amarrá-los” ao servidor – Director (comando **status client** no bconsole).
 - Simular novos jobs de backup configurados e a lista de diretórios a ser copiada (**estimate listing**).
 - Testar novos jobs de backup (e sua restauração posterior) através de um job avulso – ad hoc (comandos **run & restore**).
 - Testar o backup e o restore de aplicações específicas a serem protegidas (ex.: banco de dados, máquinas virtuais etc.).
 - Executar testes de restauração periodicamente.
- Ter uma política de backup corporativa atualizada.
- Ter um plano de recuperação de desastres e continuidade dos serviços.
- Atividade permanente de monitoração dos alertas de backup, erros e análise dos logs.
- Periodicamente conduzir auditorias de segurança.
- Prover alta disponibilidade para o serviço de backup, por ser crítico.
- Privilegiar a automação, como princípio já citado dos backups.
- Seguir as recomendações **ISO 27002**.
- Formalização dos procedimentos de backup e restauração (informação escrita sobre o que é backupado e também as solicitações de restauração).

²⁵ Alguns dados podem ser recuperados a partir de discos rígidos em ambiente de laboratório, mesmo depois de sete formatações.

4

Esquema GFS de Rotação de Backups

Grandfather-Father-Son (avô-pai-filho) é a estratégia de rotação de backup mais comum em todo o mundo. Baseia-se em backups escalonados, o que significa configuração de diferentes tempos de retenção aplicadas aos diferentes níveis de backup. Há outras estratégias de rotação de backup (por exemplo: Torre de Hanói etc.), inclusive algumas dadas por modelos matemáticos complexos, mas elas estão fora do escopo deste livro.

4.1 Backups escalonados

O oposto de backups escalonados seria usar o mesmo tempo de retenção para todos os backups executados. Isso traria uma alta granularidade de opções de restauração (por exemplo: um backup por dia), mas a retenção geral dos backups seria muito curta.

Por outro lado, se forem selecionados alguns desses jobs para terem uma retenção maior, existiriam opções de restauração de arquivos mais esparsas no passado (por exemplo: meses atrás).

A cada novo dia as mudanças diárias de um arquivo específico se tornam menos importantes, e se o usuário precisa que sejam restauradas versões diárias específicas desse arquivo é porque está trabalhando nele de alguma forma e não poderá se dar ao luxo de esperar várias semanas para solicitar uma restauração.

Isso posto, **normalmente é preferível** ser capaz de restaurar um arquivo de meses ou mesmo anos atrás, mesmo se não for a versão com alterações aplicadas em um dia específico, do que fornecer mais alguns dias de alterações de arquivos diários como opções de restauração.

4.2 Detalhes do GFS

O significado da sigla GFS é apenas uma alusão à execução de backups escalonados. Trata-se de um dos esquemas de rotação mais amigáveis porque baseia-se em fenômenos humanos corriqueiros: **dias úteis das semanas e meses**, que foram traduzidos para backups com diferentes tempos de retenção, e consequentemente **pools** (grupos de volume) distintas: **diário, semanal e mensal** (por exemplo).

A regra básica para GFS é: para começar a reciclagem de volumes para um determinado nível (backups diários), é necessário ter finalizado com êxito um backup do **nível mais alto** (backups semanais). Se não houve incidentes de perda de dados, provavelmente é seguro começar a sobrescrever volumes do backup diário da semana anterior (por exemplo: **uma fita escrita segunda-feira pode ser sobrescrita na próxima segunda-feira**).

NAM							
A							
NAMNNAI***							
Iape A							
HA							
NANMUN**							
Iape A							

28									
Backup**									
lape 1									
28									
Backup**									
lape 1									

***Backup diferencial. **F-SEMAN = Backup semanal full. ***F-MENSAL = Backup mensal full.**

Algumas empresas ainda têm uma cópia de segurança avô-pai-filho que é executada uma vez por ano (pool anual), com uma retenção ainda maior (por exemplo: **dois, três ou cinco anos**).

4.3.1 Retenções do GFS clássico

Os tempos mínimos de conservação para a rotação GFS são:

- Sete dias para os backups diários.
- Trinta dias para os backups semanais.
- Um ano para backups mensais.

Normalmente, esses valores de retenção de volume (diretiva **Volume Retention** do Bacula) são um pouco mais curtos do que o intervalo de tempo real entre os backups do mesmo nível enumerados anteriormente, de maneira a evitar sobreposição (novos backups precisam de fitas que já deveriam poder ser recicladas). Também é aconselhável definir um limitador de uso dos volumes (por exemplo: **Volume Use Duration**²⁶), para que o Bacula pare de escrever nos volumes que ainda não encheram por tempo ilimitado (o que sempre vai reiniciar a contagem do tempo de retenção dos jobs já terminados) e o volume esteja **reservado** para ser reciclado quando a retenção finalmente terminar. É possível dizer que o prazo para o volume ser reciclado seria a soma de: **Volume Use Duration + Volume Retention**²⁷.

A **Figura 4.1** a seguir é um exemplo de GFS com cinco backups por semana:

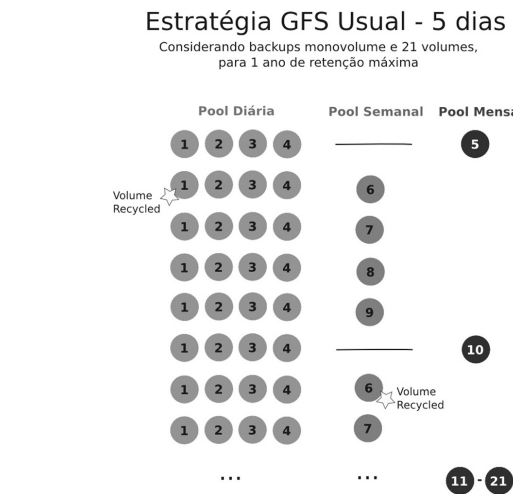


Figura 4.1. Esquema GFS usual.

4.3.2 GFS avançado apenas para fitas

Normalmente os backups diferenciais²⁸ diários são naturalmente menores que um full. E um dos fatores de desgaste do storage é a quantidade excessiva de carga e descarga (ejeção) das fitas. Dessa forma, outra opção GFS consiste em permitir ao Bacula utilizar a mesma fita para todos os **backups diários** (por exemplo, usando um maior **Volume Use Duration** – neste caso, **quatro dias**).

- Primeira semana (fita a):** se couberem, todos os backups diários ocuparão apenas esta primeira fita, que estará no drive durante toda a semana. Esta fita será descarregada automaticamente pelo Bacula apenas quando iniciar o backup semanal ou mensal daquela semana, que necessariamente vai utilizar uma fita pertencente a outra **pool** (conjunto de fitas).
- Segunda semana (fita b):** esta segunda fita é selecionada e gravada porque o **Volume Use Duration** da fita “a” já deve ter terminado, está marcada como **Used (encerrada para gravação)** e dentro do período de retenção (**Volume Retention**).
- Terceira semana (fita a):** neste momento o tempo de retenção da fita “a” já deve ter terminado e, agora, o Bacula pode reciclá-la. Ou seja: sobrescrever seu conteúdo com novos dados de backup diários.
- Quarta semana (fita b):** idem.

Este método requer apenas duas fitas para backups diários (considerando que caibam em uma mídia) e pode ser muito útil para pequenos robôs de fitas de oito slots (gavetas). A **Figura 4.2** (adiante) é uma representação gráfica deste método de trabalho alternativo:

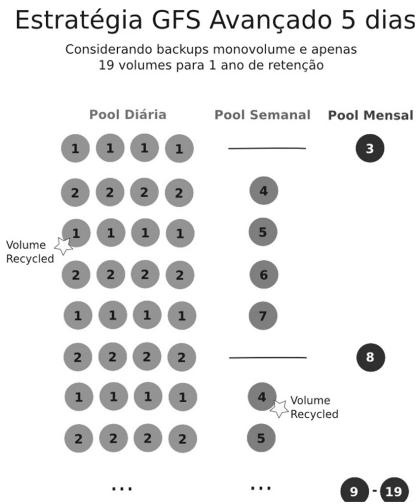


Figura 4.2. Esquema GFS avançado para fitas.

²⁶ No Bacula, **VUD (Volume Use Duration)** é o período de tempo no qual o seu volume fica aberto (ele pode receber mais dados dos jobs de backup). Começa a contar a partir do primeiro backup que está escrito no volume, e depois que termina o volume é fechado (status alterado para **used**).

²⁷ Isso será visto posteriormente no item “Ciclo de vida dos volumes do Bacula”, Capítulo 7.

²⁸ Para fitas, sempre é preferível o backup diferencial ao incremental, para evitar trocas desnecessárias de fita durante a restauração, ainda que seja um robô de fitas.

5

O Que é o Bacula?

O **Bacula** é o **terceiro** software de backup corporativo mais popular do mundo, de acordo com o serviço Google Trends. É um avançado software de backup em rede que permite backup gerenciado, indexação, restauração e verificação de arquivos, agendamento de trabalhos etc. entre diferentes servidores, independentemente dos sistemas operacionais envolvidos (multiplataforma).

5.1 Módulos do Bacula

Um sistema de backup Bacula precisa de pelo menos um dos seguintes componentes. Os daemons podem ser instalados na mesma máquina ou distribuídos em outras diferentes, de acordo com a sua finalidade.

5.1.1 Director Daemon (Diretor)

Este é o núcleo do sistema de backup e, normalmente, apenas um é realmente necessário para cada centro de dados (a não ser que haja uma quantidade incomum de clientes – ex.: mais de 4.000 ou 5.000). Ele pode ser instalado em uma máquina virtual ou física e é o responsável por receber todas as configurações de backup, tempos de retenção, agendamento, por iniciar as tarefas de backup e pela comunicação entre os demais daemons.

Na maioria dos cenários, os administradores do Bacula usam a mesma máquina do **Director** para hospedar também um Storage Daemon (armazenamento), tendo um dispositivo de disco, NAS ou robô de fitas conectados para a gravação do backup. De igual sorte e mandatoriamente, também terá instalada o **File Daemon** (cliente de backup), para backup da própria máquina. Por fim, o serviço de banco de dados (catálogo) também pode ser instalado no mesmo ambiente, embora essa distribuição não seja obrigatória.

Cada novo dispositivo de armazenamento ou cliente de backup na mesma rede que precisa ser usado no sistema de backup do Bacula deve ser, então, “anexado” ao **Director**, adicionando o respectivo recurso dentro de sua configuração (bacula-dir.conf).

5.1.2 Storage Daemon (SD)

O daemon de armazenamento é **sempre necessário** ao escrever backups do Bacula em qualquer dispositivo. Este pode ser um disco fisicamente conectado, unidades de fita, robôs de fita ou mesmo storages de disco montados através da rede, tais como **NAS – protocolo iSCSI** (NFS, CIFS etc. também são suportados, mas falta-lhes eficiência para a transferência de dados). Discos externos também são suportados, e na maioria das vezes é uma boa ideia para

fazer discos remotos ou fisicamente ligados em ponto permanente no nível do sistema operacional de montagem (por exemplo: **usando fstab**). É virtualmente possível ter **infinitos Storage Daemons** amarrados a um único **Director** (ex.: quatro robôs de fitas conectados a diferentes máquinas e escrevendo jobs de backup de diferentes clientes, em paralelo).

5.1.3 File Daemon (FD)

Este é o **cliente de backup**. Trata-se do pedaço de software que é instalado em cada máquina que precisa ter os dados copiados para fins de backup. Ele basicamente espera a conexão do **Director** para comandos como: iniciar um job de backup (ou restauração) de uma lista de diretórios ou arquivos – escrevendo (ou lendo) em um determinado **Storage Daemon**. É também responsável por processar a compressão ou criptografia via software do Bacula, se habilitados, e isso pode aumentar a carga da CPU de acordo com o nível ou as configurações usadas. O Bacula Community possui diversos pacotes distribuídos para Linux, BSD, Unix, Windows (9x, NT, 2000, XP, 2003, Windows 7, Windows 8, Windows 8.1, Windows 10, Windows Server 2012/R2, Windows Server 2016); no entanto, possui poucas dependências de software, sendo possível compilá-lo a partir do código-fonte para outros sistemas operacionais, incluindo HP-UX, Macintosh (OSX) etc.

5.1.4 Catalog (banco de dados)

Já que softwares de backup escrevem backups na forma de sequência de bytes – mais rápida e econômica –, seria muito lento listar o seu conteúdo e, eventualmente, restaurar um único arquivo dessa massa de dados sem informações adicionais. É por isso que todos os softwares de backup possuem (ainda que internamente) um banco de dados para armazenar índices do conteúdo dos volumes de backup, em que posição um trabalho começa e termina, o caminho completo para cada arquivo, metadados de arquivos etc. Dessa forma, é possível listar os arquivos a partir de qualquer backup diretamente em um console do software de backup, mesmo se o volume (por exemplo: fita) não estiver presente. Além disso, a restauração é muito mais rápida quando o Bacula sabe, aproximadamente através de índices, em qual parte do volume estão localizados os arquivos a serem lidos.

A instalação fresca do Bacula traz um job próprio de backup do catálogo configurado. Este possui um script `RunBeforeJob` que faz um dump do banco de dados e salva em um determinado diretório do sistema de arquivos que é incluído no `FileSet` e copiado pelo job. Dessa forma, é mais rápido recuperar um desastre do sistema de backup completo através da importação do dump desse banco de dados.

O Bacula suporta hoje o **PostgreSQL** e o **MySQL** para hospedar seu banco de dados de catálogo (SQLite não é mais suportado).

5.1.5 Consoles do Bacula (bconsole CLI, BAT GUI)

O console Bacula (CLI) é a forma mais básica e direta para o usuário acessar o serviço **Director** do backup, executar trabalhos, conexões de teste para os clientes e storage (status), executar restaurações etc. Ele tem muitas características e opções de fácil utilização, tais como: prompt interativo, comandos que se autocompletam, histórico de comandos, pesquisa reversa por comandos (Ctrl + R) etc.

No entanto, há uma série de outras opções de console, incluindo a BAT (Bacula Administration Tool): uma interface gráfica desktop de bibliotecas Qt que também está no código-fonte Bacula e tem a mesma sintaxe de configuração do bconsole (na verdade, você pode simplesmente copiar o conteúdo do arquivo bconsole.conf funcional para o bat.conf, a fim de definir o acesso ao Director).

5.1.6 Bacula Web Interfaces

O Bacula Community também fornece uma interface web chamada **Baculum** que permite muitas ações gráficas em vez de usar a CLI bconsole. Também existem muitos projetos de terceiros, tais como Webmin, Webacula e Bacula-web, e cada um é melhor em um aspecto, respectivamente: alterar os arquivos de configuração do Bacula; operação intuitiva e ACL para usuários; relatórios de backup. Devido a isso, muitos administradores do Bacula Community utilizam mais de uma dessas interfaces, a fim de obter as suas melhores características.

Essas interfaces web são normalmente instaladas na mesma máquina do Director, uma vez que elas geralmente precisam de acesso direto ao banco de dados de catálogo do Bacula e às vezes ao bconsole. Há um capítulo deste livro dedicado à implantação dessas interfaces.

5.2 Funcionalidades únicas do Bacula

- Formato aberto de volumes de backup (não proprietário): não há prisão tecnológica. Como último recurso, poderia ser recuperado com comandos nativos do Unix dump, tar etc²⁹.
- Suporte multibanco de dados SQL: PostgreSQL ou MySQL.
- Diferente suporte a algoritmos de compressão em software aberto: GZIP e LZO.
- Trinta anos de compatibilidade com versões anteriores de backup.
- Plug-in genérico bpipe, para automatizar e transmitir quase qualquer aplicação específica com FIFO (por exemplo: dumps de bases de dados, exportações de VMs etc.).

5.3 Principais funcionalidades do Bacula

- Backup multivolumes automático.
- Multiplexação de backup.
- Níveis de backup incremental, diferencial e full.
- Jobs de cópia, migração e verificação de integridade e confiabilidade.
- Execução de scripts antes e depois dos jobs.
- Comunicação e criptografia de dados.
- Limite de largura de banda para o job.
- Praticamente sem casos de vazamento de memória.
- Reciclagem automática dos volumes e gestão de retenções.
- Esquema de rotação de backup compatível com GFS.
- Deploy modular independente: director, armazenamento, cliente, banco de dados e consoles são diferentes daemons que podem ser executados em uma única, algumas ou milhares de máquinas.
- GPL³⁰: não há custos com licenciamento e proporciona o conhecimento do código-fonte, possibilitando o desenvolvimento de novos recursos próprios.

- Um enorme apoio da comunidade de usuários, com muitos recursos (listas de discussão, fórum, canal IRC etc.).
- Software de backup com a mais vasta literatura – inclusive através da internet, com um manual de mais de novecentas páginas.
- Alta portabilidade: Unix, Linux, Windows, MacOS etc. (desenvolvido em C e C++, podendo ser compilado para quase todos os sistemas operacionais).
- Suporte abrangente e automático de praticamente qualquer dispositivo de disco ou fitas.

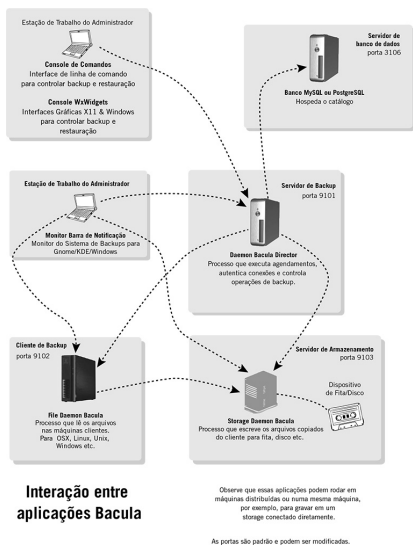


Figura 5.1. Diagrama de rede do Bacula.

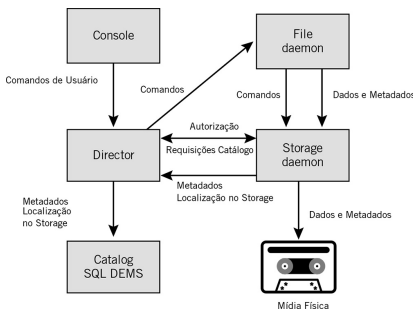


Figura 5.2. Interação entre os daemons do Bacula.

O **Bacula Community** pode ser a alternativa **perfeita** para substituir softwares proprietários como Netbackup, Arcserve etc., com mais ou menos esforço a depender de alguma aplicação específica que precise ser backupeada.

Se existirem muitas aplicações proprietárias que necessitam de procedimentos especiais de backup, a **versão Enterprise do Bacula** fornece uma ampla gama de plug-ins amigáveis que automatizam o backup e a restauração dessas aplicações específicas (VMware, Hyper-V, Xen, da Oracle, AD etc.). As licenças nunca são taxadas pelo tamanho de backup como em outros concorrentes proprietários, sendo provavelmente o software de nível empresarial de menor

custo atualmente. É desenvolvido pela empresa Bacula Systems³¹ (Suíça).

²⁹ Referência: <http://www.bacula.org/7.4.x-manuals/en/developers/Storage_Media_Output_Format.html>.

³⁰ No caso do Bacula Community. O Enterprise possui o núcleo aberto, mas licença proprietária.

³¹ <<http://www.baculasystems.com/>>.

6

Instalação do Bacula

É altamente recomendável escolher um sistema operacional atualizado **Linux de 64 bits** para hospedar o **Servidor Bacula (Director, Storage, File Daemons e banco de dados do Catálogo)**, normalmente baseado nas mais comuns **Debian e RedHat**. Pelo menos **8 GB de RAM**, **dois modernos núcleos de CPU** e **240 GB de espaço em disco rápido (considere SSD)** são especificações mínimas de hardware para um sistema decente. Lembre-se de que a exigência de espaço em disco para sistema do Bacula (não considerando os dados de backup) **pode crescer por conta de seu banco de dados**, e isso é proporcional ao número de arquivos de backup (entradas no banco de dados), ao número de jobs de backup e aos tempos de retenção dos índices no banco.

O **Bacula** pode ser seguramente executado em uma máquina virtual, mas há algumas preocupações sobre como o acesso ao seu banco de dados pode ser alguns pontos percentuais mais lento dependendo da tecnologia utilizada. Além disso, certifique-se de que é possível realizar o **pass-through** de seus dispositivos de armazenamento físicos (robôs de fitas, RDX etc.) para a máquina virtual que hospedará o Bacula Storage Daemon.

Os administradores também devem decidir se eles instalarão o **Bacula** a partir dos **pacotes já construídos** dos sistemas operacionais Linux ou se eles querem **construir a partir do código-fonte (compilação)**. Ambos os métodos têm vantagens e desvantagens expostas na **Tabela 6.1**.

Instalar pelo código	Instalar a partir de pacotes
• Ajuda a resolver problemas de compatibilidade de arquivos e pacotes e os daemons da máquina. • Diferentes versões de consistência e desempenho do Bacula. • Nem todas as versões de Linux são mais antigas do que as correções de bugs. • Não é possível usar os arquivos de configuração do Bacula para corrigir alguma configuração do sistema. • Instalação tecnicamente mais fácil (em tese). • Pode ser recomendado.	• Não é possível usar os arquivos de configuração do Bacula para corrigir alguma configuração do sistema.

Tabela 6.1. Vantagens e desvantagens dos tipos de instalação do Bacula.

Independentemente da escolha, é importante **primeiro instalar** o banco de dados que hospedará o Catálogo do Bacula, uma vez que ambos os métodos de instalação precisarão **criar o banco de dados, tabelas e o usuário bacula durante o processo de instalação**.

6.1 Instalação do servidor – banco de dados de catálogo

O Bacula foi o primeiro software de backup padrão SQL e é o único (de grandes players) a suportar **MySQL** e **PostgreSQL** na hospedagem de seu banco de dados de catálogo (SQLite não é mais suportado).

Há grandes discussões se **PostgreSQL** poderia ser mais adequado para grandes ambientes (milhares de máquinas clientes de backup e arquivos a serem copiados), mas normalmente é mais seguro escolher o que for melhor suportado na sua empresa, podendo ser o **MySQL**.

Geralmente não é necessário compilar o **SGBD** a partir do código-fonte. Ele pode ser instalado usando o seu gerenciador de pacotes favorito (apt, yum, yast etc.)

6.1.1 MySQL

Debian 8/Ubuntu etc.

Proceda com os seguintes simples comandos **apt** como usuário root Linux³²:

```
apt-get update
apt-get -y install mysql-server
```

Durante o processo de instalação, você deve ser **solicitado a definir e confirmar** a senha para o usuário administrador do MySQL (também chamado de **root**). Anote essa senha em um lugar seguro, pois vamos precisar dela mais tarde.

Após a instalação terminada, o daemon do MySQL já deve estar automaticamente em execução e configurado para carregar no momento da inicialização do servidor: pronto para receber a instalação do Bacula.

Centos 7/RedHat etc.

Proceda com os seguintes comandos **yum** como usuário root do Linux:

```
yum install -y mariadb-server
```

Inicie o serviço do MariaDB e habilite para inicialização automática em tempo de boot:

```
service mariadb start
systemctl enable mariadb
```

Agora prossiga com a instalação Bacula. A senha do usuário administrador do MariaDB é **root** e a senha estará em branco³³, informação que será necessária para procedimentos adicionais.

6.1.2 PostgreSQL

Debian 8/Ubuntu etc.

Proceda com os seguintes comandos **apt** como usuário root do Linux³⁴:

```
apt-get update
apt-get -y install postgresql
```

Após a instalação o daemon do **PostgreSQL** já deve estar em execução, configurado para iniciar automaticamente em tempo de boot e pronto para receber a instalação do Bacula.

Centos 7/RedHat etc.

Proceda com os seguintes comandos **yum** como usuário root do Linux:

```
yum install -y postgresql-server
```

Faça configuração inicial Postgresql (initdb), inicie seus daemons e registre-o para começar sempre no momento da inicialização da máquina:

```
service postgresql initdb
service postgresql start
```

```
systemctl enable postgresql
```

Prossiga com a instalação do Bacula.

6.2 Instalação do servidor – daemons do Bacula e dependências

6.2.1 Instalação por pacotes

Ao listar pacotes Bacula em seu gerenciador de pacotes **Debian** favorito, você vai notar que os pacotes **Director** e **Storage Daemons** serão diferentes de acordo com o banco de dados escolhido e instalado na última seção. Por exemplo:

```
root@hfaria-desk-i5:~# apt-cache search bacula
bacula-doc - bacula documentation
bacula - network backup service - metapackage
bacula-client - network backup service - client metapackage
bacula-common - network backup service - common support files
bacula-common-dbg - network backup service - common support files (debugging)
bacula-common-mysql - network backup service - MySQL common files
bacula-common-mysql-dbg - network backup service - MySQL common files
(debugging)
bacula-common-pgsql - network backup service - PostgreSQL common files
bacula-common-pgsql-dbg - network backup service - PostgreSQL common files
(debugging)
bacula-common-sqlite3 - network backup service - SQLite v3 common files
bacula-common-sqlite3-dbg - network backup service - SQLite v3 common files
(debugging)
bacula-console - network backup service - text console
bacula-console-dbg - network backup service - text console (debugging)
bacula-console-qt - network backup service - Bacula Administration Tool
bacula-console-qt-dbg - network backup service - Bacula Administration Tool
(debugging)
bacula-director-common - network backup service - Director common files
bacula-director-common-dbg - network backup service - Director common files
(debugging)
bacula-director-mysql - network backup service - MySQL storage for Director
bacula-director-mysql-dbg - network backup service - MySQL storage for
Director (debugging)
bacula-director-pgsql - network backup service - PostgreSQL storage for
Director
bacula-director-pgsql-dbg - network backup service - PostgreSQL storage for
Director (debugging)
bacula-director-sqlite3 - network backup service - SQLite 3 storage for
Director
bacula-director-sqlite3-dbg - network backup service - SQLite 3 storage for
Director (debugging)
bacula-fd - network backup service - file daemon
bacula-fd-dbg - network backup service - file daemon (debugging)
bacula-sd - network backup service - storage daemon
bacula-sd-dbg - network backup service - storage daemon (debugging)
bacula-sd-mysql - network backup service - MySQL SD tools
bacula-sd-mysql-dbg - network backup service - MySQL SD tools (debugging)
bacula-sd-pgsql - network backup service - PostgreSQL SD tools
bacula-sd-pgsql-dbg - network backup service - PostgreSQL SD tools
```

(debugging)

bacula-sd-sqlite3 - network backup service - SQLite 3 SD tools

bacula-sd-sqlite3-dbg - network backup service - SQLite 3 SD tools
(debugging)

bacula-server - network backup service - server metapackage

Para versões recentes do **CentOS**, os pacotes agora suportam tanto MySQL quanto PostgreSQL (padrão). Você pode selecionar qual será utilizado após a instalação com o comando **alternatives**.

Mesmo que alguns repositórios ainda exibam pacotes SQLite, eles não serão mais suportados ou desenvolvidos, portanto **não os utilize**. A seguir há algumas sugestões de conjunto de pacotes mais agradáveis e melhores³⁵:

MySQL Debian 8/Ubuntu etc.

```
apt-get -y install bacula-director-mysql bacula-sd-mysql bacula-fd bacula-console postfix mt-st mtx
```

PostgreSQL Debian 8/Ubuntu etc.

```
apt-get -y install bacula-director-pgsql bacula-sd-pgsql bacula-fd bacula-console postfix mt-st mtx
```

Durante a instalação você será solicitado a:

- **Primeiro**, configurar o **postfix**. Normalmente, basta assumir as configurações padrões sugeridas.
- **Segundo**, confirmar se deseja que o instalador de pacotes crie o banco de dados do Bacula automaticamente (recomendado).
- **Terceiro**, informar a **senha do usuário root do MySQL**.
- **Por último**, criar e confirmar uma senha para o recém-criado usuário **bacula no serviço MySQL**. Esse é o usuário padrão que o Bacula utiliza para se conectar ao seu Catálogo.

Se você tiver qualquer problema na criação do banco de dados de catálogo durante a instalação, você sempre pode tentar novamente:

```
dpkg-reconfigure bacula-director-mysql  
ou  
dpkg-reconfigure bacula-director-pgsql
```

6.2.2 Debian/Ubuntu: pós-instalação dos pacotes

Os daemons já devem estar em execução neste momento, e você deve conseguir o acesso ao prompt Bacula (**asterisco – ***) a partir do bash:

```
# bconsole  
Connecting to Director localhost:9101  
1000 OK: debian-dir Version: 7.4.4 (20 September 2016)  
Enter a period to cancel a command.  
*
```

Se os serviços não forem iniciados por algum motivo, faça isso:

```
service bacula-fd start  
service bacula-sd start
```

```
service bacula-dir start
```

MySQL e PostgreSQL Centos 7/RedHat etc.³⁶

Tenha certeza de que os repositórios Epel³⁷ estão instalados (como root):

```
yum install -y epel-release
```

Instale os pacotes do Bacula (assuma os valores padrão do Postfix se perguntado):

```
yum install -y bacula-director bacula-storage bacula-client bacula-console  
postfix mt-st mtx
```

MySQL

Execute os scripts que vão criar o banco de dados e as tabelas do Catálogo do Bacula para deixá-lo pronto para uso. Essa etapa é indispensável e é recomendado verificar as mensagens de sucesso depois de cada execução dos scripts:

```
/usr/libexec/bacula/create_mysql_database -u root -p  
/usr/libexec/bacula/make_mysql_tables -u root -p  
/usr/libexec/bacula/grant_mysql_privileges -u root -p
```

Os pacotes Bacula do CentOS, por padrão, são definidos para usar PostgreSQL. Vamos mudar isso:

```
sudo alternatives --config libbaccats.so
```

A partir do prompt, escolha **1** para escolher a biblioteca **MySQL**:

There are 3 programs which provide 'libbaccats.so'.

```
Selection  
/usr/lib64/libbaccats-mysql.so  
/usr/lib64/libbaccats-sqlite3.so  
/usr/lib64/libbaccats-postgresql.so
```

Enter to keep the current selection[+], or type selection number: 1

Prossiga para o próximo item de geração de senhas, mais adiante.

PostgreSQL

Execute os scripts de criação do banco de dados, tabelas e usuário bacula no serviço postgresql (as barras invertidas são apenas cosméticas aqui para ficar agradável no livro e indicam que os comandos continuam após a quebra de linha):

```
sudo -u postgres \  
    /usr/libexec/bacula/create_postgresql_database \  
sudo -u postgres \  
    /usr/libexec/bacula/make_postgresql_tables  
sudo -u postgres \  
    /usr/libexec/bacula/grant_postgresql_privileges
```

Prossiga.

Centos 7/RedHat etc. – Geração de senhas dos serviços do Bacula³⁸

Os empacotadores do Bacula para o CentOS decidiram não gerar automaticamente senhas iniciais aleatórias dos daemons durante a instalação, mas é obrigatório. Os comandos a seguir vão fazer isso:

```
DIR_PASSWORD=`date +%s | sha256sum | base64 | head -c 33`  
sudo sed -i "s/@@DIR_PASSWORD@@/${DIR_PASSWORD}/" /etc/bacula/*.conf
```

```
SD_PASSWORD=~date +%s | sha256sum | base64 | head -c 33~
sudo sed -i "s/@@SD_PASSWORD@@/{SD_PASSWORD}/" /etc/bacula/*.conf
FD_PASSWORD=~date +%s | sha256sum | base64 | head -c 33~
sudo sed -i "s/@@FD_PASSWORD@@/{FD_PASSWORD}/" /etc/bacula/*.conf
```

Centos 7/RedHat etc. – exceções de firewall e selinux

Adicione as exceções no arquivo de configuração do **iptables**, para os serviços do Bacula:

```
iptables -A FW-1-INPUT -m state --state NEW -m tcp -p tcp --dport 9101:9103 -
j ACCEPT
```

Ou, se regras forem gerenciadas pelo **Firewalld**:

```
firewall-cmd --permanent --zone=public --add-port=9101-9103/tcp
firewall-cmd --reload
```

Desabilite o **selinux**³⁹ temporariamente ou permanentemente após a reinicialização do sistema:

```
setenforce 0
sudo sed -i \
"s/enforcing/disabled/g" /etc/sysconfig/selinux
```

Centos 7/RedHat etc. – Configurando o início automático dos daemons do Bacula e executando pela primeira vez

Isso habilitará os daemons do Bacula para início em tempo de boot do sistema operacional:

```
systemctl enable bacula-dir
systemctl enable bacula-sd
systemctl enable bacula-fd
```

E isso vai iniciar todos os três daemons do Bacula em seu servidor de backup:

```
sudo systemctl start bacula-dir
sudo systemctl start bacula-sd
sudo systemctl start bacula-fd
```

Digitar **bconsole** como root deve levá-lo para o prompt do Bacula com sucesso!

```
# bconsole
Connecting to Director localhost:9101
1000 OK: debian-dir Version: 7.4.4 (20 September 2016)
Enter a period to cancel a command.
*
```

6.3 Construindo o servidor Bacula a partir do código-fonte (compilação)

Essas etapas pressupõem que o serviço gerenciador de banco de dados (**MySQL** ou **PostgreSQL**) já esteja instalado, como demonstrado no início deste capítulo.

6.3.1 Download do código do Bacula (é o mesmo para servidor e clientes)

Vá para a URL <<http://blog.bacula.org/source-download-center/>> e busque o arquivo mais recente (ex.: **bacula-xxx.tar.gz**) ou faça o download diretamente do meu site, além de já descompactar o código para **/usr/src/bacula***, tudo com apenas um comando:

```
wget -T 3 -qO- http://www.bacula.com.br/atual \
```

```
| tar -xvzf - -C /usr/src
```

Debian/Ubuntu – Dependências do Bacula e da compilação no Linux

MySQL:

```
apt-get install -y build-essential libreadline6-dev zlib1g-dev liblz2-dev  
mt-st mtz postfix libacl1-dev libssl-dev libmysql++-dev
```

PostgreSQL:

```
apt-get install -y build-essential libreadline6-dev zlib1g-dev liblz2-dev  
mt-st mtz postfix libacl1-dev libssl-dev postgresql-server-dev-9.4
```

Modifique as opções de acordo com a necessidade (**normalmente as últimas três, apenas para uma primeira instalação**⁴⁰).

MySQL:

```
cd /usr/src/bacula*  
./configure --with-readline=/usr/include/readline \  
--disable-conio \  
--bindir=/usr/bin \  
--sbindir=/usr/sbin \  
--with-scriptdir=/etc/bacula/scripts \  
--with-working-dir=/var/lib/bacula \  
--with-logdir=/var/log \  
--enable-smartalloc \  
--with-mysql \  
--with-archivedir=/mnt/backup \  
--with-job-email=seu@email.com \  
--with-hostname=ip_servidor_bacula_ou_nome_qualificado
```

PostgreSQL:

```
cd /usr/src/bacula*  
./configure \  
--with-readline=/usr/include/readline \  
--disable-conio \  
--bindir=/usr/bin \  
--sbindir=/usr/sbin \  
--with-scriptdir=/etc/bacula/scripts \  
--with-working-dir=/var/lib/bacula \  
--with-logdir=/var/log \  
--enable-smartalloc \  
--with-postgresql \  
--with-archivedir=/mnt/backup \  
--with-job-email=seu@email.com \  
--with-hostname=ip_servidor_bacula_ou_nome_qualificado
```

Se a configuração funcionar de acordo e não houver nenhuma dependência faltante, um sumário das opções selecionadas será impresso ao final do processo. É possível repetir o ./configure quantas vezes for necessário.

CentOS/RedHat – Dependências e compilação do servidor Bacula Linux

Adicione as exceções no arquivo de configuração do **iptables**, para os serviços do Bacula:

```
iptables -A FW-1-INPUT -m state --state NEW -m tcp -p tcp --dport 9101:9103 -  
j ACCEPT
```

Ou, se regras gerenciadas pelo **Firewallld**:

```
firewall-cmd --permanent --zone=public --add-port=9101-9103/tcp
firewall-cmd --reload
```

Desabilite o selinux⁴¹ temporariamente e permanentemente após reinicialização do sistema:

```
setenforce 0
sudo sed -i \
"s/enforcing/disabled/g" /etc/sysconfig/selinux
```

MySQL:

```
yum -y install gcc-c++ readline-devel zlib-devel lzo-devel libacl-devel mt-st
mtx postfix libssl-dev mariadb-devel
```

PostgreSQL:

```
yum -y install gcc-c++ readline-devel zlib-devel lzo-devel libacl-devel mt-st
mtx postfix libssl-dev postgresql-devel
```

Prossiga com a configuração de código, mudando as opções de acordo (**normalmente as últimas três**⁴²):

MySQL:

```
cd /usr/src/bacula*
./configure \
--with-readline=/usr/include/readline \
--disable-conio \
--bindir=/usr/bin \
--sbindir=/usr/sbin \
--with-scriptdir=/usr/libexec/bacula/ \
--with-working-dir=/var/spool/bacula/ \
--with-logdir=/var/log \
--enable-smartalloc \
--with-mysql \
--with-archivedir=/mnt/backup \
--with-job-email=seu@email.com \
--with-hostname=ip_servidor_bacula_ou_nome_qualificado
```

PostgreSQL:

```
cd /usr/src/bacula*
./configure \
--with-readline=/usr/include/readline \
--disable-conio \
--bindir=/usr/bin \
--sbindir=/usr/sbin \
--with-scriptdir=/usr/libexec/bacula/ \
--with-working-dir=/var/spool/bacula/ \
--with-logdir=/var/log \
--enable-smartalloc --with-postgresql \
--with-archivedir=/mnt/backup \
--with-job-email=seu@email.com \
--with-hostname=ip_servidor_bacula_ou_nome_qualificado
```

6.3.2 Todas as distribuições – compilar e instalar

Para construir, instalar e habilitar os daemons do Bacula para início automático em tempo de boot, respectivamente:

```
make -j8 && make install && make install-autostart
```

Configuração do banco

Execute os scripts de criação do banco de dados, tabelas e usuário do banco.

MySQL Debian/Ubuntu:

```
/etc/bacula/scripts/create_mysql_database -u root -p
/etc/bacula/scripts/make_mysql_tables -u root -p
/etc/bacula/scripts/grant_mysql_privileges -u root -p
```

MySQL CentOS/RedHat:

```
/usr/libexec/bacula/create_mysql_database -u root -p
/usr/libexec/bacula/make_mysql_tables -u root -p
/usr/libexec/bacula/grant_mysql_privileges -u root -p
```

Você será obrigado a informar a senha de root do MySQL e receberá mensagens de sucesso após cada execução de script.

PostgreSQL Debian/Ubuntu:

```
su -u postgres \
/etc/bacula/scripts/create_postgresql_database
su -u postgres \
/etc/bacula/scripts/make_postgresql_tables
su -u postgres \
/etc/bacula/scripts/grant_postgresql_privileges
```

PostgreSQL CentOS/RedHat:

```
su -u postgres \
/usr/libexec/bacula/create_postgresql_database -u root -p
su -u postgres \
/usr/libexec/bacula/make_postgresql_tables -u root -p
su -u postgres \
/usr/libexec/bacula/grant_postgresql_privileges -u root -p
```

OBSERVAÇÃO PostgreSQL: pode ser necessário também copiar os scripts anteriores para outra pasta, por exemplo: **/tmp**, de maneira a evitar erros de permissão do usuário do postgres no SO, além de dar permissão de execução para esses arquivos (por exemplo: **chmod o+rx**). Além disso, o método de autenticação padrão para o PostgreSQL pode ser um problema para a conexão do Bacula Director para essa implantação, e talvez seja necessário alterar as conexões locais do Unix **pg_hba.conf** para **trust**.

Iniciar os daemons do Bacula pela primeira vez

```
service bacula-fd start && service bacula-sd start && service bacula-dir start
```

Testar instalação do Bacula

Execute o **bconsole** a partir do Shell para acessar a CLI do Bacula (prompt indicado por asterisco **"***").

```
bconsole
Connecting to Director localhost:9101
1000 OK: debian-dir Version: 7.4.0 (16 January 2016)
Enter a period to cancel a command.
```


6.4 Instalação do Bacula e resolução de problemas no início de daemons

Se estiver trabalhando em uma nova implementação do Bacula, um dos erros mais comuns é a incapacidade do Bacula Director de se conectar ao seu banco de dados **MySQL** ou **PostgreSQL**:

1. Certifique-se de que o serviço de banco de dados está em execução.
2. Tenha certeza de que os scripts do Bacula de criação e preparação do banco foram executados com sucesso.
3. Tenha certeza de que o Bacula pode se autenticar no serviço de banco de dados (ex.: diretiva **Password**, recurso **Catalog** dentro de `bacula-dir.conf` e/ou ajuste do arquivo `pg_hba.conf` no caso de PostgreSQL). Logs do SGBD podem prover informação clara sobre erros de autenticação.

De qualquer sorte, seguem alguns comandos que irão conduzir ao **diagnóstico de problemas** do Bacula e serão úteis para a operação futura do software.

Comando de teste do Bacula

```
bacula-dir -t
```

Este comando do Shell testará a configuração do daemon do Bacula para erros de sintaxe e também para problemas de conexão com o banco de dados do catálogo. Também pode ser usado para **bacula-sd** e **bacula-fd**. Você deve reiniciar os daemons do Bacula em seguida. Se nenhum erro foi mostrado, é porque você obteve sucesso.

O log do Bacula

O arquivo de log do Bacula é escrito no arquivo definido inicialmente em **bacula-dir.conf**, recurso **Messages**. Se você seguiu o guia de instalação por compilação deste livro, ele estará em: **/var/log/bacula.log**. Ele pode indicar erros de execução de daemons e outras informações úteis. Ex.:

```
cat /var/log/bacula.log
```

O servidor de backup está com o disco cheio?

Não custa verificar, até porque muitos erros são provocados por partições cheias. No prompt do Shell:

```
df -h
```

Modo de depuração dos daemons do Bacula

Se estiver em execução, pare o daemon que você precisa diagnosticar (ex.: `service bacula-dir stop`). Em seguida, inicie-o no modo de depuração. Ex.:

```
bacula-dir -d 200
```

O número é o nível de depuração. Se mais alto o nível, mais informações da saída da depuração serão impressas na tela. Pode ser muito útil para descobrir a causa do comportamento espúrio do daemon e erros nos jobs de backup.

Problemas na execução de jobs

Para rever o log completo de um job específico, no bconsole:

```
list joblog jobid=x
```

É possível buscar a lista de todos os trabalhos executados pelo Director com outro comando do bconsole:

```
list jobs
```

Se a causa do erro ainda não estiver clara, outro recurso é executar o **Director** e principalmente o **File Daemon** no modo de depuração (explicado antes) e procurar mensagens de erro.

³² Se for uma nova instalação do Debian, você pode querer remover as linhas da configuração de repositório em DVD-ROM antes de prosseguir: **/etc/apt/sources.list**.

³³ Você pode mudar no MySQL a senha do seu usuário root com o seguinte comando: `mysqladmin -u root password 'new_password'`.

³⁴ Se for uma nova instalação do Debian, você pode querer remover as linhas da configuração de repositório em DVD-ROM antes de prosseguir: **/etc/apt/sources.list**.

³⁵ Instalar o **bacula-fd** (cliente) é importante para poder fazer backup do próprio servidor do Bacula, como se fosse qualquer outra máquina. O **Postfix** é um possível agente de transporte de correio que pode ser utilizado por bacula para retransmitir as mensagens com muito mais opções. O **mt-st** e **mtx** são necessários apenas se você já tem ou planeja armazenar backups em um **drive de fitas manual** e um **robô de fitas**, respectivamente.

³⁶ Neste momento, certifique-se de que já instalou o banco de dados desejado e seu serviço está sendo executado.

³⁷ O repositório Epel fornece opções de pacotes mais atualizados e abrangentes para o Bacula.

³⁸ Se você fizer um comando **cat** do Shell em qualquer arquivo de configuração do Bacula, vai poder observar inúmeras variáveis de senha do código do Bacula, que impedirão os daemons de inicializar, a menos que sejam substituídas. Por exemplo: **@@DIR_PASSWORD@@**.

³⁹ Se você realmente precisa do selinux, é necessário pesquisar e instalar uma política compatível com sua versão do Bacula.

⁴⁰ 1. O archive dir é o diretório padrão para armazenar backups em disco. Normalmente é escolhido o ponto de montagem para um NAS de armazenamento iSCSI, ou uma grande partição de discos do servidor local. Para backups em fita é possível ignorar essa opção e configurar um outro dispositivo depois. 2. A opção job-email, se configurada, definirá o Bacula para enviar e-mails para este endereço desde o início (pode ser necessário configurar o seu servidor de e-mail para aceitar). É possível mudar isso mais tarde em bacula-dir.conf, no entanto. 3. with-hostname é importante para definir o endereço de rede local de trabalho do servidor Bacula. Uma vez que é uma solução de rede, você não deve usar localhost ou valor equivalente, uma vez que os clientes usarão este endereço para enviar os dados.

⁴¹ Se você realmente precisa do selinux, é necessário pesquisar e instalar uma política compatível com sua versão do Bacula.

⁴² 1. O archive dir é o diretório padrão para armazenar backups em disco. Normalmente é escolhido o ponto de montagem para um NAS de armazenamento iSCSI, ou uma grande partição de discos do servidor local. Para backups em fita é possível ignorar esta opção e configurar um outro dispositivo depois. 2. A opção job-email, se configurada, definirá Bacula para enviar e-mails para este endereço (pode ser necessário configurar o seu servidor de e-mail para aceitar). É possível mudar isso mais tarde em bacula-dir.conf, no entanto. 3. with-hostname é importante para definir o endereço de rede local de trabalho do servidor Bacula. Uma vez que é uma solução de rede, você não deve usar localhost ou valor equivalente, uma vez que os clientes usarão este endereço para enviar os dados.

Configuração do Bacula

7.1 Esboço da primeira instalação do Bacula

As configurações de fábrica do Bacula nunca vão funcionar sem modificações, constituindo na verdade um risco para o ambiente de produção (por exemplo: o próprio Bacula não fazer backup dos seus arquivos de configurações em `/etc/bacula`).

A título de planejamento, esses são os **recursos** (seções da configuração do Bacula delimitados por **chaves ou braces** – “{ }”) que você provavelmente deve mudar em qualquer instalação do Bacula. Não esqueça: é uma obrigação reiniciar os daemons após cada modificação de configuração, a fim de aplicar as alterações. Além disso, você pode testar erros de sintaxe de arquivos de configuração com a opção `-t` (por exemplo: **bacula-dir -t**):

I. **Alterar a diretiva Nome do Director no arquivo bacula-dir.conf:** esta informação é importante para identificar o daemon que está enviando mensagens no log de backup, e o padrão da instalação é formulado com o nome do host da máquina. Mas essa informação também está presente nos arquivos de configuração do **Storage** e **File Daemons** e precisa ter exatamente o mesmo valor para permitir conexões de Director nesses serviços. Se quiser alterar essa informação, este é provavelmente o melhor momento, uma vez que ainda não há outros clientes remotos ou storage daemons configurados para corresponder ao valor da diretiva – apenas os locais. Consulte o item de configuração à frente para obter informações detalhadas.

II. **Nomes dos jobs são ótimos candidatos a mudanças, uma vez que essa nomenclatura é livre e vai fazer suas operações com o Bacula muito mais legais.** Em vez de “BackupClient1”, por exemplo, este será muito mais bem representado por “BkpBaculaServerConfs” ou “BackupBaculaArquivos” etc.

III. **Esquemas de rotação de backup escalonado ou GFS exigirão novas pools com nomes representativos e diferentes retenções (por exemplo: diária, semanal, mensal etc.).** Normalmente, é seguro apagar ou substituir as pools de fábrica (Default e File). A pool Scratch só é importante se você planeja usar uma biblioteca de fitas.⁴³

IV. **Altere os valores padrões das agendas (Schedule) WeeklyCycle e WeeklyCycleAfterBackup para atender às suas necessidades de frequência do backup.**

A instalação fresca do Bacula traz esta última Schedule exclusivamente para o job de backup do catálogo (para reforçar a ideia de que você poderia ter uma pool separada para hospedar esse backup do banco de dados com menores tempos de retenção), mas você poderia usar somente a agenda **WeeklyCycle** para todos os backups, e então o backup do catálogo será sempre executado com a mesma retenção de outros jobs de

backup comuns, sempre depois dos outros se for mantida sua menor prioridade (**Priority 11**). A maior funcionalidade oculta do recurso **Schedule** é que você pode personalizar todo o comportamento do backup ao longo das programações (por exemplo, diferentes pools, storages etc.) – em cada linha diferente de agenda.

V. Certifique-se de que seu dispositivo de armazenamento dos backups está gravando backups no local correto. Alguns pacotes do Bacula de diferentes distribuições Linux trazem o diretório **/tmp** como local padrão de instalação. Verifique como esse valor está configurado com o comando **status storage**, no **bconsole**, e modifique na configuração do **Storage Daemon (bacula-sd.conf)**, diretiva **Archive Device** dos **Devices** de gravação em disco (ex.: **FileStorage1 e 2**), caso necessário. Se você alterar esse caminho, também deve fornecer permissões de usuário do sistema operacional ao Bacula (por exemplo: **chown bacula /mnt/backups**).

VI. O FileSet usado pelo job de backup dos arquivos do próprio servidor do Bacula chamado “Full Set” é bastante inútil: somente copia os arquivos de **/usr/sbin**. Escolher entre fazer o backup do sistema Linux completo (por exemplo: **/**) ou apenas do essencial para restauração da aplicação instalada é uma decisão do administrador ou de política de backup. Para o backup da máquina que hospeda o Bacula, é aconselhável, pelo menos, incluir o **“/etc”** – que contém as configurações do Bacula, outros scripts e o do servidor. Os arquivos de bootstrap (pequenos arquivos que são a última recuperação de emergência do Bacula) e logs da máquina também podem ser de uso futuro (**/var**) – exceto os datafiles do banco de dados (ex.: **/var/lib/mysql**), pois este já é protegido pelo outro job de backup do banco de dados através de dump. Você também pode querer alterar “Full Set” para um nome mais significativo (por exemplo: **“bacula_files”**), lembrando que essa substituição deve ser feita em todos os recursos (jobs etc.) que chamam o antigo FileSet “Full Set”.

VII. Anexar outros clientes e eventualmente storages remotos. Depois de configurar o Director e instalar novos clientes Bacula ou storages em outras máquinas, o administrador precisa “amarrá-los” ao **bacula-dir.conf**, criando novos respectivos recursos com a respectiva senha (Password), endereço (Address) e nomes (Name) para conexão. Depois disso, teste a amarração com o comando **status client** ou **status storage** (do **bconsole**) para verificar se a conexão é possível e solucionar problemas (por exemplo: firewall do sistema operacional, erro de configuração da senha ou do nome do Director no cliente etc.). Finalmente, crie novos **FileSets e jobs** que façam backup dessas máquinas.

7.1.1 Diagrama de correspondências de valores das configurações do Bacula

As setas indicam onde os valores devem coincidir para evitar erros de autenticação de daemons do Bacula ou erros lógicos de sintaxe **bacula-dir.conf** (por exemplo: um job que especifica um uso do cliente, mas não tem o mesmo nome exato ou não existe).

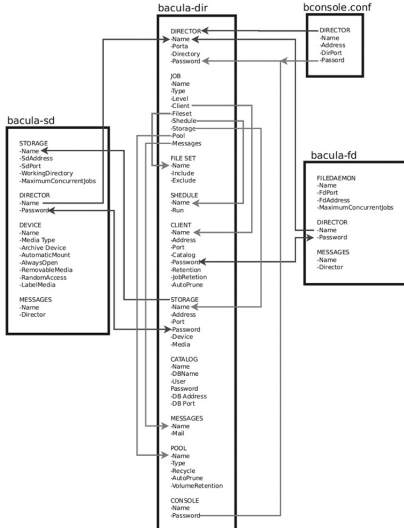


Figura 7.1. Diagrama de configuração do Bacula.

7.2 bacula-dir.conf

O Director do Bacula é o principal centro das operações de backup e não há o sistema Bacula sem ele. Ele comanda todas as outras interações de daemons e mantém a maioria das configurações de backup: jobs, schedules, FileSets, pools, clients e storages amarrados etc., sendo também o maior e mais complexo dos arquivos de configuração do Bacula.

A única razão para ter mais de um Director por empresa é se houver uma carga de trabalho realmente enorme, o que quase nunca será necessário se um **tuning** for realizado.

Já sabemos que as configurações do Bacula **são divididas em estrofes comumente conhecidas como recursos**. Esta é uma amostra de todas elas, com comentários:

Cada configuração Bacula tem um recurso que define seu daemon, por exemplo: seu nome, porta de rede na qual escuta etc.

Director {

O nome do Daemon é buscado pelo nome do host da máquina mais o sufixo do daemon (ex.: -dir). Esse sufixo é opcional, mas é recomendável mantê-lo para melhor legibilidade do log. Se você precisar alterar esse valor, DEVE modificá-lo em qualquer outra configuração de armazenamento ou daemon de arquivo, porque é uma das informações necessárias de autenticação. Se é uma nova instalação do Bacula sem daemons instalados em outras máquinas, você pode mudar isso em todos os arquivos de configuração locais com o comando do shell "sed -i s/nome_antigo-dir/nome_novo-dir/g /etc/bacula/*.conf"

Name = bacula-dir

Console padrão para diretores de porta de conexão, normalmente inalterado.
DIRport = 9101

Este é apenas um arquivo de comandos SQL que o usuário pode personalizar para que contenha suas consultas ao banco de dados do Bacula - para fins de relatórios (por exemplo) e chamadas através do comando "query" do bconsole. Se estiver vazio, você pode copiar um modelo com 20 consultas a partir do código-fonte Bacula. Por exemplo, a partir de: ../examples/sample-query.sql
QueryFile = "/etc/bacula/scripts/query.sql"

Essas diretivas são definidas na configuração da compilação e normalmente

inalteradas após a instalação.

```
WorkingDirectory = "/var/lib/bacula"
```

```
PidDirectory = "/var/run"
```

Maximum Concurrent Jobs é uma diretiva do Bacula promíscua que pode ser definida em basicamente qualquer recurso. Limita a simultaneidade de tarefas sempre no recurso onde está definida, e o padrão por omissão é "1". Se definido no Director, ele limitará os trabalhos simultâneos "globalmente" para todo sistema de backup, mesmo que a soma do Maximum Concurrent Jobs configurado em jobs, client ou storage seja maior que 20 no exemplo a seguir. Trabalhos simultâneos acima de 20 seriam enfileirados e aguardariam uma vaga deixada por um trabalho terminado para iniciar. Quanto mais jobs simultâneos, melhor.

```
Maximum Concurrent Jobs = 20
```

As senhas são outra diretriz comum do Bacula, e isso causa muita confusão. Esta é a senha dos consoles e isso pode ser confirmado pelos diagramas anteriores de rede do Bacula nos quais os consoles se conectam no Director (essa senha deve corresponder ao bconsole.conf e bat.conf). Há também um comentário à direita do valor, dizendo-lhe que esta é a senha do console. Não tem como errar!

```
Password = "NsQD3herNK2TGvoSJiJb1wZvvhb5" # Console password
```

```
Messages = Daemon
```

```
}
```

Você provavelmente irá adicionar um recurso de trabalho para cada cliente anexado ao seu Director, uma vez que isso será necessário para backupê-los. Ele precisa de muitas diretrizes para estar funcional, mas muitos de seus valores se repetem (por exemplo: tipo, nível padrão, armazenamento etc.). Em vez de especificar todas essas diretivas em cada um dos recursos job, é possível ter uma ou mais classes de tarefas com diretrizes comuns (mas se você precisar de algum valor diferente, você ainda pode alterá-lo no job ou no schedule). Este é o recurso JobDefs:

```
JobDefs {
```

```
  Name = "DefaultJob"
```

```
  # A grande maioria dos jobs configurados será de backup:
```

```
  Type = Backup
```

```
  # Este é apenas o nível padrão ao enviar backups ad hoc. Basta colocar o que  
  # você provavelmente precisará reexecutar em caso de erro (por exemplo:  
  incremental/diferencial/completo etc.)
```

```
  Level = Incremental
```

```
  # Eu pessoalmente não gosto de especificar valores de client ou FileSet  
  # aqui, uma vez que é muito provável que você terá valores diferentes para  
  # cada job de backup. Mantê-los aqui pode levá-lo a uma confusão perigosa  
  # (backup de um cliente Bacula por engano em vez de outro se você esquecer de  
  # especificar um diferente no recurso job). Esses valores precisam existir  
  # como recursos (com mesmo nome) no mesmo arquivo de configuração (bacula-  
  # dir.conf). Eu mudei o conjunto de arquivos fresco "Full Set" para um mais  
  # significativo:
```

```
  Client = bacula-fd
```

```
  FileSet = "baculafiles_set"
```

```
  # A maioria dos trabalhos será executada por uma agenda (Schedule), a menos  
  # que você tenha diferentes necessidades de retenção de backup/política/  
  # janelas. Deve existir um recurso Schedule com o mesmo nome configurado neste  
  # mesmo arquivo de configuração.
```

```
  Schedule = "WeeklyCycle"
```

```
  # Pessoalmente, prefiro não definir armazenamento aqui, mas nos recursos de  
  # pool. Dessa maneira, cria-se uma associação rígida entre pools, volumes e
```

diferentes storages (ex.: disco, fitas etc.), evitando confusão na criação de jobs de backup que poderiam ser submetidos a uma pool que não contenha volumes do storage selecionado. Ainda que isso aconteça, prevalece o storage definido na pool.

Storage = File1

Aqueles são ajustes do Bacula a fim de enviar e-mails. Este é o novo nome de recurso de configuração que existe neste mesmo arquivo, mas é possível criar perfis diferentes (por exemplo: destinatários diferentes).

Messages = Standard

Novamente, este é apenas o padrão para trabalhos ad hoc. Basta usar uma pool existente ou a que mais provavelmente terá jobs avulsos submetidos manualmente em caso de erro.

Pool = Diaria-Disco

Desde o Bacula 7 essa diretiva vem por padrão na nova configuração do Director. Ela SEMPRE deve ser utilizada (ou inserida), pois melhora o desempenho de gravação de banco de dados de backup, mesmo se você estiver usando versões anteriores.

SpoolAttributes = yes

Este é um intervalo de 1-99, onde 1 é a prioridade mais alta. Ele cria um caminho crítico entre seus backups simultâneos. Ex.: os backups de prioridade 11 só são iniciados quando todos os 10 backups são terminados.

Priority = 10

O bootstrap de gravação é um pequeno arquivo de texto com índices de jobs de backup dentro do volume. É outra opção de recuperação de desastres quando você não pode restaurar o backup do catálogo Bacula ou outra contingência, permitindo que o usuário restaure os dados de um único job de backup a partir de um volume, sem ter o Bacula funcionando (apenas o bextract). Provavelmente é uma boa ideia deixar esse valor inalterado.

Write Bootstrap = "/var/lib/bacula/%c.bsr"

Agora, o Maximum Concurrent Jobs se aplica aos próprios jobs. Cada job que usa este JobDefs pode ser executado ao mesmo tempo que outros 19 trabalhos, fazendo um total de 20. O excesso de trabalhos estará em espera. Esta diretiva e este valor não estão por padrão no Bacula, mas são aconselháveis para que não sejam um gargalo para a execução de vários jobs. Valores mais altos são melhores para o desempenho geral de backup.

Maximum Concurrent Jobs = 20

}

A configuração do trabalho é a unidade das operações do Bacula. Para cada cliente de backup instalado na sua rede, provavelmente você deseja adicionar pelo menos um recurso de trabalho de backup aqui.

Job {

Este é o primeiro job (BackupClient1) que mudei para um nome mais significativo (você pode usar sua própria nomenclatura, se desejar). Estamos usando isso para o backup dos arquivos de configuração do Bacula Server e qualquer outro conteúdo importante, exceto para o backup do catálogo (BD), que deve ter seu próprio trabalho executado posteriormente.

Name = "BackupBaculaServerFiles"

Especificar um JobDefs salva muitas linhas de configuração, já que este trabalho irá buscar suas diretivas como padrão. Entretanto, você ainda pode especificar algumas delas aqui com valores diferentes, se necessário, prevalecendo sobre diretivas iguais eventualmente existentes no JobDefs selecionado.

JobDefs = "DefaultJob"

Conforme mencionado anteriormente, prefiro definir valores de Client e FileSets no recurso de trabalho. Todas as diretivas especificadas aqui

```

substituam as definidas para o JobDefs selecionado.
Client = bacula-fd
FileSet = "baculafiles_set"
}
# Um exemplo de uma configuração de trabalho do segundo cliente. Neste caso,
um Windows.
Job {
    Name = "Backup2ndWindowsClient"
    JobDefs = "DefaultJob"
    Client = windows-fd
    FileSet = "windows_set"
}
# Este job é configurado por padrão e efetua o backup do banco de dados do
Bacula usando um script antes de executar o job, que gera um dump PostgreSQL
ou MySQL. Isso é interessante, pois outros aplicativos também podem precisar
de scripts para serem copiados corretamente.
Job {
    Name = "BackupCatalog"
    JobDefs = "DefaultJob"
    Level = Full
    # Este conjunto de arquivos copia apenas o arquivo de despejo gerado por
script. A cópia só é iniciada quando o script termina.
    FileSet="Catalog"
    # A Bacula Systems recomenda o uso de uma política de retenção separada para
o Catálogo (retenção de três dias); o que significa uma schedule e pool
diferentes para este backup. Eu, pessoalmente, gosto de ter um backup de
catálogo Bacula em cada mídia de backup, e isso seria suficiente para
restaurar um servidor de backup completo do Bacula a partir de qualquer
momento do passado que houvesse job de backup, ao custo de mais
armazenamento. Dessa forma, mudei a programação do catálogo de backup para o
mesmo backup.
    Schedule = "WeeklyCycle"
    # 1. A diretiva RunBeforeJob faz o Bacula executar o script na máquina do
Director. ClientRunBeforeJob é uma diretiva semelhante, mas executa o script
no cliente. Nesse caso, você pode usar qualquer um deles, já que tanto o
Director como o seu cliente são executados na mesma máquina. 2. Este é um
script perl, mas poderia ser um script shell, binário ou powershell, bat e
executáveis.
    RunBeforeJob = "/etc/bacula/scripts/make_catalog_backup.pl MyCatalog"
    # Este script após o trabalho deveria remover o dump gerado anteriormente,
pois neste momento já foi copiado pelo job de backup. Em particular, eu
prefiro não usar, uma vez que você pode achar útil ter o mais recente backup
do catálogo em seu disco em caso de recuperação de desastres, e você vai
precisar deste espaço para gerar novos dumps, de qualquer maneira.
    #RunAfterJob = "/etc/bacula/scripts/delete_catalog_backup"
    # O arquivo bootstrap fornece o último recurso ao recuperar um desastre do
servidor do Bacula. Ele permite restaurar um único trabalho de um volume
inteiro com o aplicativo bextract (caso não tenha mais uma instalação Bacula
em funcionamento). É um arquivo texto contendo índices dos jobs no volume.
    Write Bootstrap = "/var/lib/bacula/%n.bsr"
    # Esse job deve sempre ser executado após todos os outros backups serem
concluídos, portanto ele deve ter sempre a última prioridade (o maior
número).
    Priority = 11
}

```


O Bacula precisa de uma configuração genérica de Restore Job, mas que nunca deve ser usada através do comando `run bconsole`. Ela apenas fornece diretrizes gerais para o comando **restore** (mais adequado), por exemplo: caminho de restauração padrão (Where). Não é necessário ou recomendado criar mais de um trabalho de restauração dentro do `bacula-dir.conf`.

```
Job {  
  # Nome alterado apenas para impor uma definição genérica.  
  Name = "GenericRestoreFiles"  
  Type = Restore  
  # As configurações a seguir são apenas padrão e são substituídas automaticamente pelo comando restore ao restaurar outros backups de máquinas.  
  Client=hfaria-K46CB-fd  
  FileSet="Full Set"  
  Storage = File1  
  Pool = File  
  Messages = Standard  
  # Esse caminho de restauração padrão também funcionará para máquinas Windows. O Bacula irá restaurar no mesmo volume onde o cliente do Bacula está instalado, normalmente C:.  
  Where = /tmp/bacula-restores  
}
```

```
FileSet {  
  # Eu renomeei este FileSet (também no job que estava usando o antigo "Full Set") com a intenção de usá-lo para o backup apenas de arquivos essenciais do Bacula. Configurar um backup completo do sistema operacional ou apenas arquivos e dados essenciais de aplicativos é uma escolha muito particular da política do administrador/backup, mas o último costuma ser mais utilizado, econômico e complementar melhor backups de imagens que podem ser executados de uma máquina virtual inteira.  
  Name = "baculafiles_set"  
  Include {  
    Options {  
      signature = MD5  
      # Pode ser útil (em alguns cenários) usar compressão de software mesmo quando se utilizam fitas que já compactam dados via hardware (drive). Aquela ocorre no lado do cliente de backup, e, portanto, o tráfego de dados já estaria compactado.  
      # A presença da diretiva a seguir normalmente habilita a compressão via software e define o algoritmo (LZO ou GZIPX, onde X é o nível de compressão entre 1-9)44:  
      Compression = GZIP1  
      # Ainda dentro das opções, estes são exemplos de formas de ignorar backups específicos de tipo de arquivo usando curingas (também é possível usar expressões regulares com a diretiva RegEx). Primeiro você deve estabelecer quais arquivos do cliente Bacula irá corresponder. Por exemplo:  
      # WildFile= "*.mp3"  
      # WildFile= "*.iso"  
      # Então você diz ao Bacula para não fazer backup dos arquivos cujos nomes casem com os curingas:  
      # Exclude = yes  
    }  
    # Esta é a lista de caminhos ou arquivos que estão incluídos no backup (recursivamente). Isso é o que eu acho que é essencial para restaurar um
```

```

Bacula Director (servidor). O "/etc" contém configurações de servidor e
também as configurações do Bacula em /etc/bacula. O "/var" contém logs que
podem ser usados para auditoria e investigações, mas pode ser considerado
opcional.
File = /etc
File = /var
}
Exclude {
# Aqui deve conter os diretórios e arquivos que serão excluídos do backup.
Porque eu incluí /var, eu devo excluir (pelo menos) o diretório do banco de
dados (data files), já que estamos fazendo seu backup usando dump (catálogo
de backup).
File = /var/lib/mysql
# Essas linhas a seguir apenas seriam úteis se eu tivesse optado pelo backup
do sistema de arquivos Linux completo (ou seja, se houvesse incluído File =
/, o que não é o caso).
# File = /tmp
# File = /proc
# File = /tmp
# File = /sys
# File = /.journal
# File = /.fsck
}
}

# Um segundo exemplo mínimo de FileSet Windows.
FileSet {
Name = "windows_set"
Include {
Options {
Signature = MD5
Compression = GZIP1
# Por padrão, todos os Bacula FileSets são sensíveis a maiúsculas e
minúsculas. Se você quiser modificar isso, você pode usar a seguinte
diretiva:
# Ignore case = yes
}
# Aqui sempre use barras regulares (formato Unix) em vez das barras
invertidas do Windows. As aspas são necessárias se você tiver espaços nos
nomes das pastas.
File = "C:/Program Files/Bacula"
File = "C:/inetpub"
}
# Ao fazer o backup de um volume completo do SO do Windows, há vários
exemplos que estão listados no capítulo posterior correspondente deste
livro.
Exclude {
}
}

# Mudei a ordem padrão dos recursos de bacula-dir.conf para agrupar todos os
FileSets em sequência. Normalmente, o FileSet do catálogo apareceria depois
do schedule. Simplesmente cortei e coleí aqui.
FileSet {
Name = "Catalog"
Include {
Options {

```

```

Signature = MD5
compression = GZIP1
}
# Este é um exemplo de um backup de arquivo único. Este contém o dump do
banco de dados do Bacula, que é a saída do script "RunBeforeJob".
Normalmente não há necessidade de mudar nada aqui.
File = "/var/lib/bacula/bacula.sql"
}
}
# A schedule padrão do Bacula não traz qualquer tipo de rotação de backup
configurada (retenções diferentes = pools diferentes). A seguir, um exemplo
de agenda utilizando a estratégia GFS, com backups diários de segunda a
quinta-feira (modifique de acordo com suas necessidades):
Schedule {
    Name = "WeeklyCycle"
    Run=Differential Pool=Diaria-Disco mon-thu at 21:00
    # Aqui eu reservei a primeira sexta-feira para fazer o backup mensal. Deixar
    para a última sexta-feira do mês ou utilizar outro critério pode ser
    confuso, uma vez que os meses podem ter quatro ou cinco sextas-feiras.45 As
    demais sextas-feiras fazem o semanal, que pode ser diferencial, incremental
    ou full (de acordo com sua política de backup, gastando mais ou menos
    armazenamento).
    Run=Differential Pool=Semanal-Fita 2nd-5th fri at 21:00
    Run=Full Pool=Mensal-Fita 1st fri at 21:00
}
# Desde que eu optei por usar uma única programação também para o backup de
catálogo, eu posso descartar a segunda agenda:
# Schedule {
# Name = "WeeklyCycleAfterBackup"
# Run = Full sun-sat at 23:10
# }
# Todo cliente Bacula instalado na sua rede deve ser amarrado aqui. Depois de
criar um novo recurso Client similar ao que se segue, é muito provável que
você queira criar também um novo job para fazer backup desse cliente, e lá
você especifica o uso desse Client fazendo referência ao valor da diretiva
Name.
Client {
    Name = bacula-fd
    # O endereço IP do servidor Bacula ou nome qualificado:
    Address = 192.168.1.51
    FDPort = 9102
    Catalog = MyCatalog
    # Deve corresponder ao valor desta senha do cliente bacula-fd.conf.
    Password = "linux_client_password"
    # Essas retenções não afetam os dados de backup escritos, mas apenas
    entradas na base de dados do Bacula (evitar o crescimento excessivo do
    Catálogo). Isso é detalhado posteriormente em um item de capítulo
    específico, mas eu recomendo sempre aumentar esses volumes e apenas reduzi-
    los se necessário, e com ciência das consequências.46
    File Retention = 1 year
    Job Retention = 1 year
    # Quando a retenção de file/job tiver expirado, o Bacula automaticamente
    limpará as entradas do banco de dados expiradas e provavelmente manterá seu
    catálogo com um tamanho menor.47
    AutoPrune = yes

```

Segunda amostra de cliente, neste caso Windows:

```
Client {  
    Name = windows-fd  
    # O endereço IP da máquina cliente ou nome qualificado:  
    Address = 192.168.1.60  
    FdPort = 9102  
    Catalog = MyCatalog  
    # O Password deve corresponder ao valor da senha do cliente bacula-fd.conf.  
    Password = "windows_client_password"  
    File Retention = 1 year  
    Job Retention = 1 year  
    AutoPrune = yes  
}
```

Aqui nós amarramos os dispositivos dos storage daemons da nossa rede ao Director, para que ele possa dizer aos clientes que escrevam backups neles. A nova instalação do Bacula vem com dois armazenamentos de disco, por isso é fácil de configurar jobs de cópia de um para outro. Antes de fazer qualquer alteração a seguir, no entanto, você provavelmente vai querer modificar ambos os pontos de montagem de gravação definidos em ArchiveDevice para discos locais definitivos com grande capacidade ou um NAS (FSTAB), excluir um deles ou adicionar uma nova configuração para robôs de fitas, tudo isso posteriormente no arquivo (bacula-sd.conf).

```
Storage {  
    # Este é apenas o nome que cada dispositivo é apresentado no Bacula e  
    bconsole. Você pode modificá-los, apenas renomeando-o também em outros  
    pedaços de configurações aqui em bacula-dir.conf que usam esse valor  
    (normalmente job e JobDefs).  
    Name = File1  
    # O Bacula é uma aplicação distribuída em rede, por isso não faz sentido  
    dizer aos clientes para escrever no endereço "localhost"! Certifique-se de  
    inserir aqui um IP de rede local (ou nome qualificado) que os clientes  
    possam acessar através da porta 9103 quando um job de backup for iniciado.  
    Address = 192.168.1.51  
    SDPort = 9103  
    # Cada daemon de armazenamento em sua rede pode ter uma senha diferente,  
    geralmente gerada aleatoriamente quando eles são instalados. Apenas  
    certifique-se de que este valor coincide com a senha na lista de acesso do  
    Director em bacula-sd.conf, ao amarrar um novo dispositivo de storage daemon  
    aqui.  
    Password = "storage_password"  
    # Isto DEVE coincidir com o Autochanger (grupo de dispositivos) ou um único  
    Device (apenas no caso de drive de fitas manual) em bacula-sd.conf.  
    Device = FileChgr1  
    # Isto DEVE coincidir com o Autochanger (grupo de dispositivos) ou um único  
    Device em bacula-sd.conf. Este não é um nome reservado e não afeta qualquer  
    outro comportamento dos dispositivos de escrita, mas serve para o  
    administrador estabelecer uma matriz de compatibilidade entre diferentes  
    dispositivos (volumes que pertencem a um armazenamento ou outro).  
    Media Type = File1  
    # Este "Maximum Concurrent Jobs" limita a quantidade máxima de jobs  
    simultâneos que podem ser gravados para esse dispositivo do storage  
    especificamente. Normalmente, quanto mais, melhor (tempo total dos backups  
    mais rápidos).  
    Maximum Concurrent Jobs = 10
```

```

}
Storage {
    Name = File2
    Address = 192.168.1.51
    SDPort = 9103
    # Todos os dispositivos na mesma instalação Storage Daemon terão a mesma
    senha de conexão do Director.
    Password = "storage_password"
    Device = FileChgr2
    Media Type = File2
    Maximum Concurrent Jobs = 10
}

```

Um exemplo de anexo de biblioteca de fitas conectada ao mesmo storage daemon dos dispositivos anteriores (mesma máquina do Director).

```

Storage {
    Name = IBMLibrary
    Address = 192.168.1.51
    SDPort = 9103
    Password = "storage_password"
    Device = LTO-4
    Media Type = LTO-4
    Maximum Concurrent Jobs = 20
    # Apenas para certificar-se de que o Director fará alterações de fita
    automaticamente conforme necessário.
    Autochanger = yes
}

```

É assim que o Bacula se conecta ao seu banco de dados do catálogo, e seu sucesso é necessário para que o daemon do Director (bacula-dir) seja iniciado. Você poderia ter um segundo recurso de catálogo para usar dois catálogos (por exemplo) em vez de um, mas este seria um cenário muito raro e incomum, necessitando de um balanceamento de carga manual entre os bancos de dados. Entretanto, apenas um catálogo bem tunado dificilmente será um gargalo de backup.

```

Catalog {
    Name = MyCatalog
    # Você pode querer mudar o usuário do banco de dados bacula no MySQL ou
    PostgreSQL se estiver em branco, em seguida, substituí-lo aqui (db_password)
    para uma maior segurança. É possível usar outra diretiva (dbaddress) se o
    serviço de banco de dados estiver em execução em outra máquina de rede.
    dbname = "bacula"; dbuser = "bacula"; dbpassword = "db_password"
}

```

O recurso Messages permite que o Bacula envie informações importantes como: logs de backup e status de término (mailcommand); os alertas de ação necessários para que os trabalhos paralisados continuem - disco de storage cheio; solicitar a inserção de fitas que não estão no robô etc. - são enviados para os endereços definidos pela diretiva operatorcommand.

```

Messages {
    Name = Standard
    # Isto é altamente customizável e é provavelmente uma boa ideia instalar um
    serviço Mail Transport Agent local como o Postfix para fornecer mais opções
    de envio de mensagens (por exemplo, envio de e-mail seguro autenticado
    (TLS), uma vez que a maioria dos serviços de correio e firewall bloqueia a
    porta SMTP padrão: 25). Você também pode encaminhar mensagens para o
    WhatsApp ou Telegram (veja mais em <http://www.bacula.com.br>).
}

```

```

mailcommand = "/usr/sbin/bsmtp -h localhost -f \"\\(Bacula\\) \\<%r\\>\" -s
\"Bacula: %t %e of %c %l\" %r"
operatorcommand = "/usr/sbin/bsmtp -h localhost -f \"\\(Bacula\\) \\<%r\\>\" -s
\"Bacula: Intervention needed for %j\" %r"
# Se você criou o Bacula a partir do código-fonte, você poderia ter
especificado seu endereço de e-mail com a opção -with-job-email no ./
configure. Se não, você pode apenas mudar a seguir. Vários endereços podem
ser especificados apenas repetindo as linhas.
#Mail = heitor@bacula.com.br = all, !skipped
MailOnError = heitorfaria@gmail.com = all, !skipped

MailOnSuccess = heitorfaria@gmail.com = all, !skipped
operator = heitor@bacula.com.br = mount
console = all, !skipped, !saved
# Aqui é onde o Bacula salva todos os logs dos jobs localmente. Ele também
os armazena em seu banco de dados do catálogo.
append = "/var/log/bacula.log" = all, !skipped
# Isso diz ao Bacula para armazenar logs também no banco de dados
(catálogo).
catalog = all
}
Messages {
# Essas são mensagens menos importantes da operação dos daemons (por
exemplo: não é possível conectar-se a outras mensagens do daemon etc.), mas
também é interessante recebê-las. Basta informar o endereço de e-mail
necessário na diretiva de e-mail.
Name = Daemon
mailcommand = "/usr/sbin/bsmtp -h localhost -f \"\\(Bacula\\) \\<%r\\>\" -s
\"Bacula daemon message\" %r"
mail = heitor@bacula.com.br = all, !skipped
console = all, !skipped, !saved
append = "/var/log/bacula.log" = all, !skipped
}
# As pools podem ser usadas para segregar jobs de backup com as mesmas
propriedades, como tempos de retenção. Elas permitem implantar rotação de
dados de backup e backups escalonados. Recomenda-se criar pelo menos uma pool
para diferentes mídias (fita, dispositivos de disco etc.), uma vez que devem
ter propriedades diferentes. Eu gosto de excluir as pools Default e File da
configuração fresca do Bacula e criar outras com nomes mais significativos.48
Pool {
# Criar um nome significativo único para a sua pool.
Name = Diaria-Disco
# Atualmente, não há nenhum outro tipo de pool implementada no código do
Bacula que não seja "Backup".
Pool Type = Backup
Recycle = yes
AutoPrune = yes
# Isso é importante se você quiser se certificar de que a retenção de volume
começará a ser contada e que será possível prever o exato dia e horário em
que um volume poderá ser reciclado caso haja necessidade. No caso a seguir,
20 horas + 6 dias após o início do primeiro backup (aproximadamente uma
semana). 20 horas de "Volume Use Duration" significa que o mesmo volume será
mantido aberto e gravado desde o início dos backups, permitindo que você
chegue ao trabalho na parte da manhã e execute novamente algum job
eventualmente terminado com falha, podendo ainda ser gravado no mesmo
volume.

```

```

Volume Use Duration = 20 hours
# Um volume escrito na segunda-feira deve ser capaz de ser reciclado na
próxima. Este é um valor mínimo para o esquema de rotação de backup do GFS
clássico, uma vez que você deve ter um trabalho de hierarquia mais alto
(semanal ou mensal) terminado em uma semana, mas que pode ser aumentado para
mais opções de restauração diárias (granularidade) ao custo de usar mais
espaço de armazenamento.
Volume Retention = 6 days
# É importante limitar o uso de backup de disco porque a reciclagem do
volume é uma ação atômica (tudo ou nada). Volumes maiores significam que
você teria que descartar lotes de dados de backup maiores para reciclar
(sobrescrever) um volume gravado antigo. O tamanho dele é alocado e cresce
dinamicamente quando backups são gravados, e não há nenhuma razão lógica
para usar essa diretiva para fitas magnéticas, na medida que em só é
possível ter um volume por mídia.
Maximum Volume Bytes = 50G
# Esta diretiva fornece belos e significativos nomes para os volumes de
disco (arquivos) que o Bacula deverá criar automaticamente em seu storage de
disco. Ex.: Diaria-Disco-0, Diaria-Disco-1 etc. Você pode customizar essa
sintaxe incluindo outras variáveis do Bacula, mas não há necessidade real de
usar as de data (pois o banco de dados Bacula já grava essa informação
quando um volume é criado, que pode ser visualizado com o comando llist
media no bconsole).
Label Format = "${Pool}-${NumVols}"
# Especificar o storage aqui evita erros de operador ou de configuração ao
submeter jobs. Ele cria uma matriz forte entre pools e storages,
certificando-se de que um backup executado para esta pool sempre use esse
storage (nunca outro, cujos volumes não estejam nessa pool, mesmo que
selecionados pelo operador por equívoco).
Storage = File1
}

Pool {
# Esta pool eu criarei para backups de fita.
Name = Semanal-Fita
Pool Type = Backup
Recycle = yes
AutoPrune = yes
# Três dias é um bom exemplo se seus backups semanais começam na noite de
sexta-feira. Ele permite que o administrador chegue na manhã de segunda-
feira e execute novamente os trabalhos com falha na mesma fita.
Volume Use Duration = 3 days
# Uma fita gravada na segunda sexta-feira do mês deve ser capaz de ser
reciclada no próximo mês, também a partir da segunda sexta-feira. Este é um
valor mínimo para o esquema de rotação de backup do GFS clássico, uma vez
que você deve ter um job de hierarquia mais alto (mensalmente) terminado
dentro de um mês, mas pode ser aumentado para mais opções de restauração ao
custo de usar mais mídia.
Volume Retention = 30 days
# Isso é importante para garantir que o Bacula não tente escrever em fitas
de limpeza.
Cleaning Prefix = "CLN"
Storage = IBMLibrary
}

Pool {

```

```

Name = Mensal-Fita
Pool Type = Backup
Recycle = yes
AutoPrune = yes
# Três dias é um bom exemplo se seus backups semanais começam na noite de
# sexta-feira. Ele permite que o administrador chegue na manhã de segunda-
# feira e execute novamente os trabalhos com falha na mesma fita.
Volume Use Duration = 3 days
# Uma fita escrita na primeira sexta-feira de janeiro deve ser capaz de ser
# reciclada na primeira sexta-feira de janeiro do próximo ano. Esta será a
# retenção máxima, dada pela sua política de backup.
Volume Retention = 360 days
Cleaning Prefix = "CLN"
Storage = IBMLibrary
}
# Scratch é um nome reservado para pool do Bacula e fornece um poder
# especial: atribuir volumes a outras pools se elas exigirem novos volumes para
# gravar backups. Isso é muito útil, pois, ao criar rótulos para várias novas
# fitas em um robô de fitas, só é dada a escolha para utilizar apenas uma pool.
# E seria muito demorado prever e distribuir fitas em diferentes pools de
# backup. Colocar todas as novas fitas na pool Scratch automatiza esse
# processo.
Pool {
    Name = Scratch
    Pool Type = Backup
}
# Esta configuração de console é opcional e é seguro excluí-la. Útil somente
# se você quiser instalar o aplicativo Bacula Tray Monitor, que provê
# mecanismos de monitoração do backup na estação de trabalho do administrador
# de backup.
# Console {
# Name = bacula-mon
# Password = "BU7RBbBdoY8Jy7kZmlAmK8feTMp/83rv/B2YdHPP3Z74"
# CommandACL = status, .status
#}
##### Fim do bacula-dir.conf #####

```

7.3 bacula-sd.conf

O bacula-sd (**Bacula Storage Daemon**) é o banco de sangue dessa organização de vampiros (armazena os dados de backup). Ele sempre vai fazê-lo em uma forma sequencial (mesmo se escrevendo em discos rígidos), porque é mais rápido e requer menos espaço do que a replicação de um sistema de arquivos (este não é o propósito do Bacula).

Os volumes podem receber dados de backup multiplexados de clientes de diferentes sistemas operacionais; compactados ou não; criptografados ou não. O Bacula também pode executar backups em vários volumes (caso a capacidade de um deles se esgote) e restaurá-los automaticamente, tanto para volumes em discos ou fitas (backup mono ou multivolume).

O administrador precisa instalar o bacula-sd em cada máquina que tem um dispositivo de armazenamento (hardware) fisicamente conectado e quer escrever backups nele.

Uma máquina para o **Storage Daemon** pode eventualmente ser diferente daquela que hospeda o **Director do Bacula**, mas normalmente é instalado na mesma. Também é possível

ter vários **Storage Daemons** espalhados em sua rede, mas conectados ao mesmo **Director**, então seria possível escolher quais jobs de backup seriam gravados em cada um.

Neste exemplo de **bacula-sd.conf** a seguir, considere que está instalado na mesma máquina do **Director** mostrado no último tópico. Além disso, contém três tipos de dispositivos utilizados: **(1) um conjunto de backups de disco com dois discos** (configuração fresca do Bacula SD para backups em disco/ponto de montagem); **(2) uma biblioteca de fitas com duas unidades**; **(3) uma única unidade de fita manual**.

```
Storage {
# O nome do daemon de armazenamento é composto pelo nome do host da máquina
com o sufixo -sd na instalação nova. O sufixo -sd é opcional, mas é muito
útil para distinguir o proprietário das mensagens de erro do log Bacula. É
possível mudar esse nome sem qualquer impacto para outra operação do Daemon
Bacula, no entanto. Ele não é usado para fins de autenticação como o nome do
Director bacula-dir.conf).
Name = bacula-sd
SdPort = 9103
WorkingDirectory = "/var/lib/bacula"
PidDirectory = "/var/run"
# Esse limite de MaximumConcurrentJobs afeta a soma de todos os dispositivos
configurados neste daemon. Somente se este for acima deste máximo, fará com
que os trabalhos excedentes esperem por um slot de execução.
Maximum Concurrent Jobs = 50
}
# Esse recurso permite que o Director se conecte a este daemon de
armazenamento. É raro, mas possível, que mais de um Director se conecte ao
mesmo daemon de armazenamento, provavelmente configurado para escrita em
diferentes dispositivos/AutoChanger. Neste caso, basta duplicar a estrofe e
configurá-la corretamente.
Director {
# É obrigatório definir aqui o nome correto do seu Director Bacula, conforme
configurado em bacula-dir.conf, recurso Director.
Name = bacula-dir
# Senha para a conexão com o Director com todos os dispositivos configurados
neste daemon de armazenamento. Deve ser o mesmo que é inserido no recurso de
armazenamento (bacula-dir.conf) ao amarrar esses dispositivos ao Director.
Password = "storage_password"
}
# A estrofe a seguir tem um "Monitor = yes", então não faça uma confusão com
o recurso anterior do Director. Essa configuração do monitor não é
obrigatória e pode ser comentada/excluída se você não usar o Bacula Tray
Monitor.
# Director {
#     Name = bacula-dir
#     Password = "monitor_password"
#     Monitor = yes
# }
# Esta é uma configuração de armazenamento em disco. Em vez de usar um
dispositivo para cada ponto de montagem, é mais prático agrupá-los e usar um
recurso AutoChanger como a configuração do Bacula fresca a partir da versão
5. Você pode criar mais de um dispositivo para o mesmo ponto de montagem, o
que geralmente leva a um melhor desempenho devido a um balanceamento de carga
entre os dispositivos (ainda que utilizem o mesmo ponto de montagem de
disco).
```

```

Autochanger {
# Este nome pode ser alterado a critério, mas deve corresponder ao valor da
diretiva Device no recurso Storage do bacula-dir.conf.
Name = FileChgr1
# Listar aqui os dispositivos virtuais (devices) que o Bacula usará para
este AutoChanger. Eles devem ser configurados individualmente, logo a
seguir.
Device = FileChgr1-Dev1, FileChgr1-Dev2
# Este comando está vazio porque estamos lidando com um disco, LVM ou
storage que estão sempre montados no Linux e nenhum comando é necessário
para alternar entre mídia em dispositivos diferentes.
Changer Command = /dev/null
Changer Device = /dev/null
}

Device {
Name = FileChgr1-Dev1
# O MediaType não é um nome reservado, portanto o Bacula não usa essas
informações para alterar qualquer parâmetro de desempenho de escrita, mas é
absolutamente importante informar que os volumes criados neste armazenamento
pertencem a ele. Em outras palavras: cada dispositivo diferente no bacula-
sd.conf que grave em locais diferentes deve ter valor de MediaType
diferente.
Media Type = File1
# Os discos devem ser configurados (normalmente no fstab) para serem sempre
montados durante o boot, para que o Bacula possa acessá-los. O usuário
bacula do Linux precisará de permissões completas para este ponto de
montagem.
Archive Device = /mnt/backup
# Permite que o Bacula crie volumes automaticamente. Normalmente, o desejado
para backups de disco, como as outras diretivas a seguir.
LabelMedia = yes
Random Access = Yes
AutomaticMount = yes
RemovableMedia = no
AlwaysOpen = no
# Faz com que o Bacula balanceie a carga entre os dois dispositivos. Se
houver mais de cinco trabalhos simultâneos sendo executados, o Bacula vai
começar a usar o próximo dispositivo de disco também.
Maximum Concurrent Jobs = 5
}

Device {
Name = FileChgr1-Dev2
Media Type = File1
Archive Device = /mnt/backup
LabelMedia = yes
Random Access = Yes
AutomaticMount = yes
RemovableMedia = no
AlwaysOpen = no
Maximum Concurrent Jobs = 5
}

# Este é um exemplo de configuração de biblioteca de fitas. É semelhante ao
disco, pois também consiste num grupo de dispositivos de escrita e leitura
(AutoChanger).

```

```

AutoChanger {
# Inicialmente, este nome pode ser alterado a critério, mas deve
corresponder ao valor da diretiva Device no recurso de armazenamento bacula-
dir.conf quando amarrado ao Director.
Name = IBMLibrary
# Listar aqui os dispositivos virtuais que o Bacula usará para este
AutoChanger de acordo com quantas unidades sua biblioteca de fitas tiver.
Device = Drive-1
Device = Drive-2
# Normalmente não é necessário mudar isso. É o script/comando que muda a
fita automaticamente.
Changer Command = "/etc/bacula/scripts/mtx-changer %c %o %S %a %d"
# Este deve ser o caminho estático para o dispositivo de biblioteca de fita
(braço do robô, picker etc.). O caminho estático para o dispositivo pode ser
descoberto no shell com o comando: "ls -la /dev/tape/by-id/*", que é
preferível aos caminhos genéricos/dinâmicos (ex.: /dev/sgx) porque eles
podem ser alterados quando o servidor é reinicializado.
Changer Device = /dev/tape/by-id/scsi-SSTK_L700_XYZZY_A
}
Device {
Name = Drive-1
# Drive Index deve começar a partir de 0 e ser incrementado. Esta ordem de
unidades deve corresponder à ordem física deles na biblioteca, que pode ser
descoberta no shell com o comando "mtx -f /dev/changer_device status".
Drive Index = 0
# O MediaType não é um nome reservado, portanto Bacula não usa essas
informações para alterar qualquer parâmetro de desempenho de escrita, mas é
absolutamente importante informar que os volumes criados neste armazenamento
pertencem a ele ou são compatíveis entre si (por exemplo: se podem ser lidos
em uma unidade de fita manual sobressalente).
Media Type = LTO
# A seguir, deve ser o caminho estático para o dispositivo de unidade de
fita. Pode ser descoberto no shell com o comando: "ls -la / dev / tape / by-
id / *". É SEMPRE preferível usar a nst (forma não rebobinável), que é mais
adequada para a operação do Bacula. Use caminhos estáticos em vez de
genéricos /dev/stx porque eles normalmente alteram a ordem quando o servidor
é reinicializado.
Archive Device = /dev/tape/by-id/scsi-35033441ab1000-nst
# As próximas configurações de diretrizes são tipicamente as melhores para
bibliotecas de fitas. Você também pode querer adicionar alguns parâmetros de
tecnologia de fita específicos que podem ser estudados no capítulo
específico do livro sobre Tuning.
AutomaticMount = yes
AlwaysOpen = yes
RemovableMedia = yes
RandomAccess = no
AutoChanger = yes
Alert Command = "sh -c 'tapeinfo -f %c |grep TapeAlert|cat'"
Alert Command = "sh -c 'smartctl -H -l error %c'"
# Faz com que o Bacula balanceie a carga dos jobs entre os dois
dispositivos. Se houver mais de cinco trabalhos simultâneos sendo
executados, o Bacula começará a usar a próxima unidade de fita também.
Maximum Concurrent Jobs = 5
}
Device {

```

```

Name = Drive-2
Drive Index = 1
Media Type = LTO
Archive Device = /dev/tape/by-id/scsi-350224ab00200-nst
AutomaticMount = yes
AlwaysOpen = yes
RemovableMedia = yes
RandomAccess = no
AutoChanger = yes
Alert Command = "sh -c 'tapeinfo -f %c |grep TapeAlert|cat'"
Alert Command = "sh -c 'smartctl -H -l error %c'"
Maximum Concurrent Jobs = 5
}

```

Este é um exemplo manual de configuração da unidade de fita. Diferentemente dos exemplos comentados do bacula-sd.conf, a sugestão a seguir requer menos intervenção manual, por exemplo: requerer um comando umount do bconsole/eject das fitas depois de estarem cheias. De qualquer forma, é necessário montar a próxima fita depois de inseri-la. Você pode criar um script RunBeforeJob para montar a fita antes do primeiro job de backup agendado de um dia específico (consulte o capítulo de comandos).

```

Device {
    Name=ManualTapeDrive
    # A seguir, deve ser o caminho estático para o dispositivo de unidade de
    # fita. Pode ser descoberto no shell com o comando: "ls -la /dev/tape/by-id/
    # *". É SEMPRE preferível usar a nst (forma não rebobinável), que é mais
    # adequada para a operação do Bacula. Use caminhos estáticos em vez de
    # genéricos /dev/stx porque eles normalmente alteram a ordem quando o servidor
    # é reinicializado.
    Archive Device = /dev/tape/by-id/scsi-36450ab00200-nst
    Changer Device = /dev/null
    # Responsável pela automação manual da unidade de fita: ligeiramente
    # diferente da configuração real de uma biblioteca de fitas.
    Changer Command = "/etc/bacula/scripts/mtx-changer %o %a %S"
    # Opcional: apenas para não deixar o Bacula à espera de uma troca de fita.
    Maximum Changer Wait = 3d
    # Se eu usar o mesmo tipo de mídia usado na configuração anterior da
    # biblioteca de fitas, o Bacula pensará que eles são compatíveis entre si. Eu
    # também preciso usar ao menos duas pools diferentes para segregar as fitas da
    # biblioteca e desta unidade manual, a fim de evitar confusão. Se a biblioteca
    # de fitas e as unidades manuais não forem compatíveis, basta usar um tipo de
    # mídia diferente.
    Media Type = LTO
    # Isso é necessário para que o Bacula não tente inserir outra fita depois
    # que a anterior for ejetada.
    Offline On Unmount = Yes
    # As próximas configurações de diretrizes são tipicamente as melhores para
    # unidades de fita. Você também pode querer adicionar alguns parâmetros de
    # tecnologia de fita específicos que podem ser estudados no capítulo do livro
    # sobre Tuning.
    AutomaticMount = yes
    AlwaysOpen = yes
    RemovableMedia = yes
    # Use esta diretiva mais o "LabelFormat" no recurso pool (bacula-dir.conf)
    # se você quiser que o Bacula crie automaticamente volumes para as fitas.
    Label media = Yes
}

```

```

}
Messages {
  Name = Standard
  # A diretiva informa o cliente para enviar suas mensagens de funcionamento
  do daemon para o Director com esse nome, para que ele possa retransmitir
  para o seu e-mail etc.
  Director = bacula-dir = all
}
### Fim do bacula-sd.conf ###

```

7.4 bacula-fd.conf

O **bacula-fd** é o próprio **vampiro**: deve ser instalado em todas as máquinas que você precisa que o Bacula copie os arquivos e os envie para o Storage Daemon (**ele suga o sangue/dados dos computadores!**). Durante a restauração os dados percorrem o caminho inverso, entretanto.

Em algumas situações, os administradores de backup pensam em exportar sistemas de arquivos para uma máquina com o Bacula File Daemon instalado em vez de instalá-lo na máquina que deseja fazer cópia do sistema de arquivos. Isso deve ser evitado a todo custo, uma vez que esses protocolos de exportação (**CIFS, NFS, SMB, FTP etc.**) geralmente possuem graves limitações de desempenho, pois funcionam na camada sete do modelo OSI (aplicação).

O arquivo de configuração bacula-fd.conf está normalmente presente no diretório /etc/bacula para distribuições Linux, e para Windows pode ser acessado através do **Menu Iniciar > Todos os Programas > Bacula > Configuration > Edit Client Configuration** (link para: **C:\Arquivos de Programas\Bacula**). Tanto no Windows como no Linux o arquivo bacula-fd.conf possui os mesmos recursos e são similares, exceto para a especificação de caminhos dentro do sistema operacional.

Os pontos básicos que normalmente são alterados neste arquivo de configuração são: o **nome real do seu Director** e uma **senha** para seu acesso, já que sem essa informação correta o **Director** não poderá acessá-lo depois que **amarrarmos** o cliente (ou seja, **adicionarmos um novo recurso Client correspondente no bacula-dir.conf**).⁴⁹ A ordem dos recursos do bacula-fd.conf também pode variar em instalações frescas, mas isso não causa nenhum impacto.

Também é possível alterar o nome (diretiva Name) do **Bacula File Daemon** se você achar que é melhor do que o criado por padrão. Veja um exemplo:

```

FileDaemon {
  # Nome do cliente normalmente pode ser alterado da maneira desejada.
  Name = bacula-fd
  # A porta normalmente é padrão e inalterada.
  FdPort = 9102
  WorkingDirectory = "/var/lib/bacula"
  PidDirectory = "/var/run"
  # Permite que o Bacula execute várias tarefas de backup e restauração
  simultaneamente para este mesmo cliente se o storage e o Director não
  estiverem limitados pela mesma configuração. Normalmente é o desejado.
  MaximumConcurrentJobs = 15
}

Director {

```

```
# Deve necessariamente ter o valor igual ao nome do Director definido em
bacula-dir.conf (servidor de backup).
Name = bacula-dir

# Em uma nova instalação, é possível alterar essa senha, se desejado. De
qualquer sorte, copie essa informação (por exemplo: em um bloco de notas),
porque ela deve ser inserida no bacula-dir.conf ao anexar esse cliente
(adicionar um novo recurso Client).
Password = "linux_client_password"
}

# Esta configuração do monitor (Monitor = sim) só é útil se você tiver
instalado uma ferramenta gráfica chamada Bacula Tray Monitor, que poderia ser
instalada em qualquer máquina, mas faz mais sentido se você tiver GUI nos
sistemas operacionais do servidor. Como este segundo recurso com nome
Director pode causar confusão, é seguro comentar ou excluir essa configuração
se não pretende utilizar esta funcionalidade.
#Director {
#   Name = bacula-dir
#   Password = "mon_password"
#   Monitor = yes
#}
Messages {
    Name = Standard
    # Isso informa o cliente para enviar suas mensagens de backup importantes
    para o Director, para que ele possa retransmitir para o seu e-mail etc.
    Director = bacula-dir = all
}
### Fim do bacula-fd.conf ###
```

7.5 bconsole.conf (bat.conf)

A última das configurações típicas do Bacula não afeta um daemon, mas uma aplicação que permite o acesso ao sistema de backup para controle direto de um usuário – o administrador do Bacula.

O **bconsole** é a CLI e o **BAT** (Bacula Administration Tool, se instalado ou compilado) é a GUI nativa desktop de administração do Bacula. Assim, o **bconsole.conf** e o **bat.conf** (respectivamente) podem ter basicamente as mesmas configurações para um mesmo usuário acessar o sistema.

Os consoles acessam somente o **Bacula Director** por padrão na porta 9101 e é mais seguro instalá-los e usá-los na estação de trabalho **Linux** ou **Windows** do administrador, a fim de evitar fazer o acesso terminal remoto ao servidor de backup (ssh etc.).

Esta é uma configuração simples padrão bconsole.conf (ou bat.conf):

```
Director {
    # Ele deve corresponder ao Nome do Director definido no diretório bacula-
    dir.conf.
    Name = bacula-dir
    # O endereço IP da máquina ou nome qualificado.
    Address = 192.168.1.51
    DirPort = 9101
    # Deve corresponder à senha do Director do bacula-dir.conf (provavelmente o
```

```
primeiro do seu arquivo).  
Password = "console_password"  
}
```

Observação: é possível ter mais de um recurso “Director” e usar o mesmo console para se conectar a directors diferentes, se for o caso, de maneira alternada.

7.6 Novos clientes Bacula

Os clientes do Bacula não têm muitas dependências, e, como eles são desenvolvidos em C/C++, eles podem ter seus binários construídos basicamente em qualquer sistema operacional. É importante lembrar que a versão do cliente **não pode ser maior** do que a do **Director** e **Storage Daemons**, embora o oposto possa acontecer.

É importante certificar-se de que seu cliente seja compilado com suporte a backup de ACL, compressão GZIP e/ou LZO, suporte de criptografia (se pretende algum dia utilizar essas funcionalidades), e isso é feito através da instalação das respectivas bibliotecas de desenvolvimento adequadas. Veremos a seguir algumas possibilidades de instalação apenas de clientes.

7.6.1 Clientes Linux

Pacotes

A maioria das distribuições traz binários do Bacula totalmente funcionais em seus repositórios estáveis por padrão e sua instalação é quase sem esforço. Por exemplo:

```
apt-get install bacula-fd
```

ou talvez:

```
yum install bacula-client
```

para distribuições CentOS/RedHat/Fedora (etc.).

Após a instalação é hora de configurar o novo cliente, explicado no final desta seção.

Compilação apenas do cliente

Download do código-fonte do Bacula (mesmo para servidor e clientes)

Vá para o URL <<http://blog.bacula.org/source-download-center/>> e pegue o último arquivo bacula-xxx source.tar.gz ou simplesmente baixe diretamente a versão 7.4 do meu site, descompacte e acesse o diretório (/usr/src/bacula*) com os comandos a seguir⁵⁰:

```
wget -T 3 -qO- http://www.bacula.com.br/Atual \
| tar -xzf - -C /usr/src
```

Prossiga para instalar as dependências de pacotes necessários.

Debian 8/Ubuntu⁵¹:

```
apt-get install -y build-essential zlib1g-dev liblzo2-dev libacl1-dev  
cd /usr/src/bacula*  
./configure \
--enable-client-only \
--enable-build-dird=no \
```

```
--enable-build-stored=no \  
--bindir=/usr/bin \  
--sbindir=/usr/sbin \  
--with-scriptdir=/etc/bacula/scripts \  
--with-working-dir=/var/lib/bacula \  
--with-logdir=/var/log \  
--enable-smartalloc
```

CentOS 7/RedHat:

Adicione exceções adequadas necessárias para o Bacula para iptables/Firewalld:

```
firewall-cmd --permanent --zone=public --add-port=9102/tcp  
firewall-cmd --reload
```

Desative o selinux e reinicie, ou pesquise e aplique uma política para o Bacula.

```
sudo sed -i "s/enforcing/disabled/g" /etc/sysconfig/selinux  
reboot
```

Continue com as dependências e configure:

```
yum -y install gcc-c++ zlib-devel lzo-devel libacl-devel  
cd /usr/src/bacula*  
./configure \  
--enable-client-only \  
--enable-build-dird=no \  
--enable-build-stored=no \  
--bindir=/usr/bin \  
--sbindir=/usr/sbin \  
--with-scriptdir=/usr/libexec/bacula/ \  
--with-working-dir=/var/spool/bacula/ \  
--with-logdir=/var/log \  
--enable-smartalloc
```

Todas as distribuições

Crie, instale e ative o daemon bacula-fd no início da máquina:

```
make -j8 && make install && make install-autostart-fd
```

Inicie o daemon do cliente do Bacula pela primeira vez. Por exemplo:

```
service bacula-fd start
```

Proceda com a configuração do cliente e amarração ao Director, como explicado no item 7.6.3.

7.6.2 Instalação de clientes Windows

Já existem pacotes construídos no site do Bacula no Source Forge. Eles contêm o **cliente**, **Storage Daemon** e as **consoles bconsole** e **BAT** para Windows, mas para este procedimento estamos **apenas instalando o cliente**, uma vez que é mais usual para backup de servidores de produção.

- Baixe e descompacte os pacotes dos clientes do meu site²: <http://www.bacula.com.br/winclients>.
- Execute a instalação como administrador (do sistema e do domínio, se aplicável). Aceite o contrato de licença.

- Selecione o arquivo .exe que deve atender a sua arquitetura do Windows: **32 ou 64 bits**. Ex.: **bacula-win64-7.4.0.exe**.

- Selecione **Instalação Automática** (personalizada é usada apenas para scripts de implantação automática mais avançados que podem incluir instalação silenciosa).
- Escolha o software que deseja instalar (neste exemplo, apenas o cliente Windows), como na **Figura 7.2**.

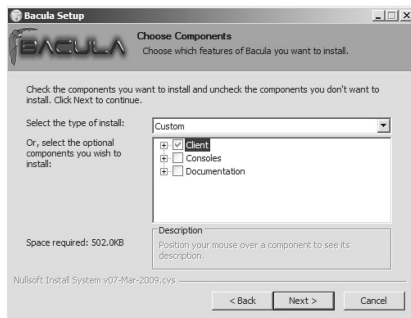


Figura 7.2. Seleção de componentes do pacote do Windows do Bacula.

- Preencha o nome do Director conforme configurado em sua máquina do servidor Bacula Director (**bacula-dir.conf – recurso Director**). Você também pode obter essa informação ao acessar o **bconsole** ou usando o comando **version**.

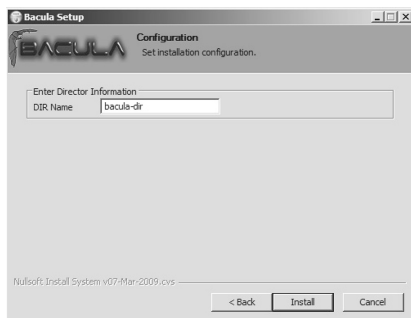


Figura 7.3. Solicita o nome real do Director.

- Aguarde o final da instalação. Feche os prompts/telas adicionais.
- Prossiga com a configuração do cliente.

7.6.3 Configuração dos clientes Bacula

bacula-fd.conf (lado do cliente)

- Edite o arquivo de configuração do cliente:
 - Linux: `vi /etc/bacula/bacula-fd.conf`.
 - Windows: Menu Iniciar > Bacula > Configuration > Edit Client Configuration.
- Edite/certifique-se de que o primeiro recurso de nome **Director (aquele sem a diretiva Monitor = yes)**⁵³ contém exatamente o mesmo nome configurado na sua máquina do servidor do **Bacula Director (bacula-dir.conf – recurso Director)**.⁵⁴ Você também pode obter essa informação ao acessar o **bconsole** ou usando o comando **version**.

•Copie a senha gerada aleatoriamente ainda no mesmo recurso Director do bacula-fd.conf (**diretiva password**) ou altere-a conforme desejar. Você precisará dessa informação mais tarde ao anexar esse cliente ao Director.

•Altere a diretiva **Name** do **FileDaemon** (recurso **FileDaemon**) se julgar que não é adequado ou que este cliente pode ser um nome mais significativo.

•Salvar e sair. Reinicie o serviço do cliente:

–Linux: /etc/init.d/bacula-fd restart.

–Windows: acesse o Gerenciador de Serviços (services.msc) e reinicie o serviço Bacula File Daemon.⁵⁵

•Verifique se o sistema operacional possui um firewall ativo. Se sim, adicione uma exceção **TCP 9102** de entrada para que o Director possa se conectar a esse cliente. Ou execute o seguinte comando no terminal do Windows:

```
netsh advfirewall firewall add rule name="bacula-fd" dir=in action=allow protocol=TCP localport=9102
```

bacula-dir.conf (lado do servidor)

•Adicione um novo recurso de cliente (você pode copiar e colar de um já existente e alterar os valores das diretivas necessárias). Altere o nome do cliente (**Name**) para o mesmo nome configurado em bacula-fd.conf (este realmente não precisa ser o mesmo, mas penso que seja uma boa prática); em **Address** coloque o endereço para o **IP** ou **FQDN** (nome qualificado) de sua máquina cliente; e, por fim, em **Password**, coloque a senha que foi copiada anteriormente também do bacula-fd.conf. Por exemplo:

```
Client {
    Name = bacula-fd
    Address = 192.168.1.199
    FDPort = 9102
    Catalog = MyCatalog
    Password = "xos2o70KCHf4namn9vs5fENUGNuEf5WaxWtL8j"
    File Retention = 1 year
    Job Retention = 1 year
    AutoPrune = yes
}
```

•Salve e saia do arquivo **bacula-dir.conf**.

•Reinicie o **bacula-dir** ou recarregue-o no bconsole para aplicar as alterações (comando **reload**).

•No **bconsole**, faça um **status client** para o novo cliente amarrado. Se ele retornar o nome do cliente, versão e lista de jobs, isso significa que sua configuração foi um sucesso e o **Director pode se conectar ao seu cliente**.

Agora que o novo **client** foi anexado ao **Director** e testado, é hora de criar um **novo job** e um **novo FileSet**. Volte para editar o **bacula-dir.conf** mais uma vez:

•Adicione um novo recurso job (sugestão: copie um trabalho de backup do Bacula já existente para facilitar a digitação).

•Renomeie o job para um nome distinto significativo. Este novo job deve especificar o mesmo nome do cliente que foi usado no recurso do cliente. Aproveite essa oportunidade para atribuir um **FileSet** diferente e que ainda não existe, mas que estaremos criando com o mesmo nome logo adiante.

```
Job {
```

```
Name = "BackupLibreOffice"
JobDefs = "DefaultJob"
Client = libreoffice-fd
FileSet = fileserver_set
}
```

•Crie um novo FileSet com o mesmo nome especificado no job (por exemplo: **FileSet = fileserver_set**). Lembre-se de alterar o **Include** e **Exclude** com as pastas que você precisa ter no backup do novo cliente. Lembre também que para os diretórios do Windows você deve usar **barras regulares** e nunca **barras invertidas**, e que caminhos com **espaços no nome** devem ser escapados com aspas. Por exemplo:

```
File = "C:/Program Files"
```

•Reinicie o bacula-dir ou recarregue-o com o comando reload no bconsole para aplicar as alterações.

•Teste suas novas configurações com o comando **estimate listing** do bconsole. Ele imprimirá os arquivos que o Bacula iria fazer backup caso você iniciasse um job, além do somatório do tamanho e da quantidade dos arquivos que seriam copiados.

•Submeta seu primeiro job de backup (**run**), se desejado. Não esqueça de acompanhar a execução do trabalho lendo suas mensagens (comando **message** do bconsole), usando os comandos **status director**, **client** ou **storage** (cada um fornece um detalhe diferente do job em execução) ou usando o **list jobs**, que fornece um **resumo codificado** dos trabalhos em execução e terminados (consulte os últimos anexos deste livro para obter o significado dos códigos).

7.7 Envio de e-mail do Bacula

O Bacula usa (de fábrica) um aplicativo próprio para envio de e-mail: o bsmtpt. Dependendo das suas configurações de serviço de correio, é possível que o Bacula envie e-mails diretamente para ele, substituindo o endereço do **Mail** e **Operator Command** no recurso de mensagens (**bacula-dir.conf**):

```
mailcommand = "/usr/sbin/bsmtpt -h localhost -f \"\ (Bacula\ ) <%r>\" -s  
\"Bacula: %t %e of %c %l\" %r"
```

Neste caso, **localhost** poderia ser substituído, por exemplo, por **mail.bacula.com.br**.

O **MailCommand** envia mensagens sobre backups terminados, juntamente com logs de backup. As variáveis %t %e %c %l são usadas para compor o assunto do e-mail: tipo de job, status de término, client e nível do job, respectivamente. Isso pode ser personalizado.

O **OperatorCommand** envia mensagens sobre a intervenção humana necessária para os jobs do Bacula, pois eles provavelmente estão paralisados. Ex.: trocar fitas da unidade ou biblioteca de fitas, forçar a reciclagem de volumes, fornecer mais espaço de armazenamento em disco para escrita de backup etc.

Ainda no recurso **Messages**, é especificado o endereço de e-mail que receberá as mensagens, **Mail** e **Operator**.

```
mail = heitor@bacula.com.br = all, !skipped  
operator = heitor@bacula.com.br = mount
```

É possível repetir ambas as linhas se você precisar especificar mais de um receptor, ou mesmo um alias Linux para mais de um endereço.

Observação: você pode enfrentar limitações enviando mensagens diretamente do `bsmtp` (por exemplo: impossibilidade de fornecer autenticação de e-mail, bloqueio de tráfego na porta 25, falta de suporte SSL etc.). Devido a isso também é possível instalar um serviço de mensagens locais (por exemplo: Postfix) e até mesmo fornecer segurança aprimorada para envio de e-mail (a seguir).

7.7.1 Postfix e envio de e-mail autenticado

Se você instalar o Postfix na máquina do servidor do Bacula, o recurso de mensagens **bacula-dir.conf** deve redirecionar as mensagens para **localhost** (padrão):

```
mailcommand = "/usr/sbin/bsmtp -h localhost -f \"\" \"(Bacula)\" \"<%r\>\" -s  
\"Bacula: %t %e of %c %l\" \" %r\""
```

O serviço **Postfix** pode ser facilmente instalado através do seu gerenciador de pacotes Linux e funciona com as configurações padrões. Por exemplo:

```
apt-get install postfix
```

Quando solicitado, deixe o “Site Internet” (padrão) marcado e pressione OK (**Figura 7.4**).

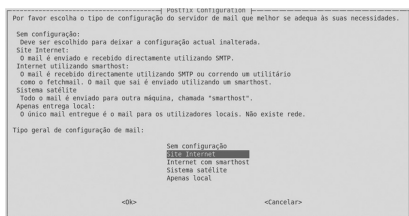


Figura 7.4. Basta escolher a configuração padrão do Postfix.

Continue selecionando os valores padrões nas próximas telas até que a instalação termine.

Para testar o envio de e-mail, você pode executar o **MailCommand** no shell ou **executar um job de backup** do Bacula e aguardar seu término. Lembre-se de verificar se as mensagens não estão chegando ao seu diretório de correio como **spam**.

Se você ainda não estiver recebendo mensagens, verifique os seguintes logs para erros – ex.: **/var/log/mail.log** e **/var/log/mail.err**.

Se instalar o Postfix não é suficiente para enviar mensagens através do seu serviço de correio ou se pretende fornecer mais segurança ao envio de mensagens, prossiga com essas definições (exemplo do Gmail, mas funciona com qualquer serviço).

Edite o arquivo de configuração postfix:

```
vi /etc/postfix/main.cf
```

Adicione as seguintes linhas (substitua uma diretiva **relayhost** se já existente):

```
smtp_sasl_auth_enable = yes  
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd  
smtp_sasl_security_options = noanonymous  
smtp_use_tls = yes  
relayhost = [smtp.gmail.com]:587
```

Salvar e sair. Em seguida, edite/crie este outro arquivo:

```
vi /etc/postfix/sasl_passwd
```

Adicione a seguinte linha. Substitua em conformidade o endereço do serviço de correio, o endereço da conta de e-mail e a senha da conta de e-mail.

```
[smtp.gmail.com]:587 email@gmail.com:password
```

Salvar e sair. Execute os seguintes comandos para aplicar as alterações:

```
postmap /etc/postfix/sasl_passwd
postfix reload
```

Teste o envio do e-mail. Se desejar, pode apagar o arquivo sasl_passwd anteriormente criado para não deixar suas credenciais disponíveis.

7.8 ACL de consoles (contas de usuário)

O acesso ao console de fábrica do Bacula é feito com um usuário sem nome e direitos completos de administrador. Isso pode ser considerado inseguro por algumas políticas de segurança e limitaria as possibilidades de disseminar o uso de Bacula para outros usuários (por exemplo: equipe de monitoramento, autoatendimento do usuário etc.).

Felizmente, é possível **criar credenciais de usuários** com papéis diferentes no Bacula (por exemplo: somente restauração, backups de um cliente específico etc.). Essas limitações de credenciais funcionarão não só com o bconsole da CLI, mas também com algumas GUIs, como BAT.⁵⁶

Exemplo de bconsole.conf (ou bat.conf)

Truncar a senha antiga do Director e adicionar um recurso de console com as credenciais de usuário:

```
Director {
  Name = bacula-dir
  Address = 192.168.10.51
  Password = "XXX" # Isso deve estar errado de propósito (não deve coincidir
    com o valor da Diretiva Password do recurso Director)
}

Console {
  Name = nome_usuario
  Password = "senha_usuario"
}
```

bacula-dir.conf

Crie um novo recurso de console com as mesmas credenciais de usuário usadas no bconsole.conf e com os direitos que o usuário deve ter:

```
Console {
  Name = nome_usuario
  Password = "senha_usuario"
  JobACL = "BackupJob"
  ClientACL = bacula-fd
  StorageACL = File
  ScheduleACL = *all*
  PoolACL = *all*
```

```
FileSetACL = "Full Set"  
CatalogACL = MyCatalog  
CommandACL = run  
}
```

Opções de ACL

```
JobACL = [lista de jobs]
```

Quais tarefas o usuário poderá executar, listadas por nomes de jobs separados por vírgulas.

Por exemplo:

```
JobACL = "Backup client 1", "Backup client 2" ...  
JobACL = "RestoreFiles"
```

A mesma sintaxe pode ser usada para as outras ACLs:

```
ClientACL = [lista nome dos clientes]  
StorageACL = ...  
ScheduleACL = ...  
PoolACL = ...  
FileSetACL = ...  
CatalogACL = ...  
CommandACL = ...  
WhereACL = ...
```

Observação 1: todos os objetos do Bacula não especificados na lista serão proibidos ao respectivo usuário.

Observação 2: para todas as ACLs é possível permitir acesso a tudo com a palavra-chave ***all***.

7.9 Usando includes para dividir arquivos de configuração do Bacula

Em vez de ter todos os recursos de configurações do Bacula dentro de um mesmo arquivo, é possível ler o conteúdo de seções diferentes espalhadas em arquivos diferentes. Esses são alguns exemplos de "includes" no bacula-dir.conf:

```
@/etc/bacula/bacula-dir-filesets.conf  
@/etc/bacula/bacula-dir-jobs.conf  
@/etc/bacula/bacula-dir-jobdefs.conf  
@/etc/bacula/bacula-dir-clients.conf  
@/etc/bacula/bacula-dir-storage.conf  
@/etc/bacula/bacula-dir-pools.conf  
@/etc/bacula/bacula-dir-schedules.conf
```

Alguns administradores também gostam de criar um arquivo diferente para cada Bacula Client, que conteria os recursos de configuração Client, FileSet e Job para essa máquina específica.

Observação: apesar de parecer tentador ter uma configuração cheia de "includes", pode ser mais difícil identificar erros (as mensagens podem não ser muito claras) e é mais complexo para encontrar uma estrofe específica. Além disso, torna-se impossível usar alguma interface gráfica que permita alterar as configurações graficamente (por exemplo:

7.10 Ciclo de vida dos volumes do Bacula

Ao trazer novas mídias (fitas ou discos) para o sistema de backup do Bacula, é necessário criar volumes antes de começar a escrever novos backups. Isso é comumente conhecido como a operação **label** (rotular).

O primeiro status de volume é sempre **append**, o que significa que ele está pronto para receber informações de backup de um ou mais jobs.

Se outro limitador não estiver definido, esse volume receberá dados de backup dos jobs agendados para a pool (conjunto) a qual pertence até obter o status **Full** (cheio). Em outras palavras: o Bacula sempre vai preferir escrever no último volume escrito, dentro da mesma pool, desde que ainda seja gravável (tenha espaço). E somente quando o volume é fechado para escrita é que podemos garantir que o temporizador do **volume retention** não será reiniciado mais, e que depois de percorrido o tempo definido ele tenha a possibilidade de ser **reciclado** (sobrescrito).

Dessa forma, com as **configurações de fábrica do Bacula** não é possível prever quando o volume vai ser reciclado, uma vez que o tamanho do volume e os tamanhos dos backups podem ser muito diferentes, para menos ou para mais. Isso pode incomodar alguns administradores que querem ter certeza de que serão capazes de reciclar um volume específico em um determinado momento.

Outro fator que influencia esse ciclo de vida é que algumas empresas possuem políticas sobre ter que usar mídia de backup diferente todos os dias ou com outra frequência. É por isso que temos que aprender os diferentes métodos de reciclagem automática e limitadores de uso de volume que o Bacula pode fornecer.

7.10.1 Reciclagem bruta

A seguinte diretiva não é recomendada e é bastante perigosa, principalmente se não souber o que ela faz. **Purge Oldest Volume = yes sempre reciclará o volume mais antigo de uma pool** toda vez que o Bacula precisar gravar seus backups, **mesmo que eles estejam dentro do tempo de retenção (volume retention)**.

O fato de essa reciclagem não respeitar a retenção de volume é tão crítico que se você tiver um tamanho de backup maior do que o tamanho do seu volume na pool haverá um **deadlock**: um mesmo volume pode ser escrito várias vezes para um mesmo job de backup e ele vai acabar fatalmente em erro.

A opção que habilita esse comportamento agressivo seria definida no recurso da pool (bacula-dir.conf):

```
Purge Oldest Volume = Yes
```

Essa reciclagem bruta poderia ser útil se escrever novos backups prontamente (sem a necessidade de aprovisionar mais fitas ou espaço em disco) for mais importante do que

guardar os volumes para essa retenção configurada, mas exigiria uma administração bastante proativa para garantir que o tamanho dos backups somados não cresça além das suas capacidades de volumes em uma mesma pool.

Felizmente, os métodos a seguir são muito mais seguros e mais adequados para ambientes de produção.

7.10.2. Reciclagem por tempos de retenção

```
VolumeRetention = xx
```

Retenção é o conceito de backup mais importante e é comum a todos os softwares. No entanto, é crucial entender como isso acontece e que é possível ter outras retenções diferentes para informações relacionadas a backup (por exemplo: metadados armazenados no catálogo – banco de dados).

O **volume retention** é o intervalo de tempo no qual o software de backup não pode descartar ou substituir automaticamente as informações gravadas, a menos que haja uma intervenção humana explícita desse modo (comando **purge**). Isso é importante também para que os volumes possam ser reciclados de forma automática após a sua retenção ter expirado, evitando erros humanos que vêm com operações manuais.

Como a retenção do volume é o tempo mínimo no qual os dados são preservados e a **reciclagem de volume** é uma ação **atômica** (tudo ou nada), o temporizador de retenção só pode ser contabilizado quando nenhum novo backup for gravado no mesmo volume. Isso pode acontecer quando o volume fica cheio (**full**) ou quando se limita o uso do volume, seja pelo **tempo no qual fica aberto para gravação**, **número de jobs que podem ser gravados em um mesmo volume**, **tamanho do volume** etc. Observe que mais de um desses limitadores pode ser utilizado ao mesmo tempo.

```
Use Volume Once = yes
```

O Bacula deveria usar apenas o volume uma vez, mas essa implementação não é a melhor nem recomendável. Pode ser equivalente e mais seguro usar **MaximumVolumeJobs = 1** (explicado adiante).

```
Maximum Volume Jobs 5= nnnn
```

Essa diretiva limita o número de jobs que você executa em um mesmo volume, o que pode ser demasiado complexo se você deseja criar um volume por dia (por exemplo), já que este número variará ao longo do tempo. No entanto, pode ser utilizado quando os administradores querem forçar que o Bacula crie um volume por tarefa ao fazer backups baseados em disco (**MaximumVolumeJobs=1**, mesmo não havendo benefícios muito claros nessa segregação dos dados em mais volumes).

```
Volume Use Duration = ttt
```

Este é meu favorito. Consiste no intervalo de tempo que começa a partir da primeira gravação de backup no volume. Depois que o **VolumeUseDuration** for excedido, o volume será fechado (o status do volume será modificado automaticamente para **used**) e **não será possível gravar novos jobs de backup no volume**, certificando-se de que a contagem de tempo de **volume retention** prossiga e não seja resetada. É útil, também, usar um valor de **VolumeUseDuration** que permita ao administrador chegar ao trabalho no dia seguinte e ser

capaz de executar novamente tarefas com falha para o mesmo volume (ex.: **20 horas**).

VolumeUseDuration pode ser utilizado se também se deseja limitar a quantidade de tempo que uma determinada mídia (tipicamente fita) permanecerá sendo escrita mesmo que ela ainda não tenha ficado cheia (ex.: **menos de 24 horas, forçando o robô a gravar em uma fita nova todos os dias**).

Maximum Volume Bytes = mmm

Este limitador **não deve** ser usado para fitas, mas é **necessário** para backups gravados em disco. Ele fecha o volume assim que seu tamanho coincide com o **MaximumVolumeBytes** especificado (o status do volume muda para **full**) e pode ser usado em simultaneidade com os outros limitadores. Seu objetivo, no entanto, é um pouco diferente: serve para fornecer granularidade para a reciclagem de espaço em disco, uma vez que a limpeza de volume para sobrescrita é uma ação atômica (tudo ou nada). Isso será visto mais detalhadamente no capítulo sobre **dispositivos de armazenamento em disco**.

7.10.3 Exemplos de retenção de volumes (GFS)

Backups diários

Suponhamos que você deseje alcançar uma retenção de **sete dias** e ter certeza de que um volume ou fita gravado em uma segunda-feira pode ser sobrescrito (caso necessário) na próxima segunda:

```
Pool {  
    Name = Diaria  
    ...  
    Volume Use Duration = 18 hours  
    Volume Retention = 6 Days  
}
```

Dessa forma, após **seis dias** e mais **18 horas** a partir do início de um determinado backup, seu volume será capaz de ser reciclado. É importante dizer que o Bacula não descarta de imediato as informações de backup escritas nos volumes depois desse período, já que sempre tenta preservar as informações ao máximo. O volume só é efetivamente sobrescrito quando há novos backups que reivindicam espaço para gravação. Assim, é seguro usar valores menores do que o esperado para a estratégia GFS clássica (**por exemplo, sete dias para backups diários**).

Observação: mesmo que o volume possa ser reciclado, pode não estar em um dado momento, pois o Bacula sempre irá preferir usar **volumes novos (recém-criados ou anteriormente reciclados)** se eles estiverem na mesma pool do job de backup enviado.

Backups semanais

Se você iniciar backups semanais **sexta à noite**, você pode ter uma maior janela de backup e não precisar iniciar novos backups no **sábado e domingo**, dependendo da sua política de backup e das necessidades do negócio. Isso é o que este exemplo cobre, sendo capaz de executar novamente na segunda-feira se o backup falhar. Para este exemplo, um volume gravado na terceira sexta-feira do mês pode ser reciclado na terceira sexta-feira do mês seguinte:

```
Pool {
  Name = Semanal
  ...
  Volume Retention = 30 Days
  Volume Use Duration = 70 hours
}
```

Backups mensais

É muito provável (e recomendável) que ao menos os seus backups mensais sejam **full**, e também poderíamos usar a mesma dinâmica do backup semanal para a duração de utilização do volume (por exemplo, um volume de backup de janeiro poderia ser reciclado em janeiro do ano seguinte):

```
Pool {
  Name = Mensal
  ...
  Volume Retention = 362 Days
  Volume Use Duration = 70 hours
}
```

7.10.4 Implantação da estratégia GFS

Uma configuração básica do Bacula GFS pode ser feita através das seguintes direções.

Pelo menos três pools distintas (bacula-dir.conf)

Pools “Diária”⁵⁷, “Semanal”, “Mensal” etc., lembrando que é aconselhável usar diferentes pools para storages diferentes (por exemplo: Diária-Disco, Diária-Fitas etc.), e esses nomes de pools não são reservados (você pode usar sua criatividade).

Claro que você pode precisar alterar isso de acordo com a sua política de backup corporativo. Por exemplo 1: se você tiver uma retenção total de um mês apenas, provavelmente não há sentido ter uma pool mensal, mas somente backups diários e semanais. Por exemplo 2: se você deve implementar uma retenção de backup total superior a um ano, é possível criar uma pool anual, mantendo esse backup somente por vários anos.

Observação: se você realmente quer que o Bacula escreva em um volume específico para uma determinada data, você deve criar uma pool para cada um (ex.: Diaria-segunda-feira, Semanal-2a-sexta-feira etc.). Essa é uma abordagem mais manual e só pode ter qualquer vantagem se você trabalhar com unidades de fita manuais.

GFS schedule (agenda)

O Bacula é o lugar para definir todo o seu comportamento de backup ao longo do tempo. É possível especificar não apenas pools diferentes para um determinado job, mas também **cada aspecto do comportamento de determinado backup** (por exemplo: todas as diretivas do **recurso job** incluindo a **pool**, **level**, **NextPool** etc.). Essas diretrizes de programação prevalecem sobre as diretivas padrão definidas no **recurso Job** (bacula-dir.conf).

```
Schedule {
Name = agenda_gfs
Non-Difference Mon-Thu
Non-Difference 2nd-5th Friday
```

```
Run15Mensual 1st Friday
```

```
}
```

Neste exemplo, espera-se que haja backups diários diferenciais de segunda a quinta-feira; backups diferenciais semanais da segunda até a quinta sexta-feira do mês (se acontecer); e backups mensais a cada primeira sexta-feira do mês.⁵⁸

7.10.5 Outras retenções do Bacula

O **volume retention** que estudamos não se relaciona com a retenção de informações do catálogo do Bacula (jobs e retenção de arquivos) que visa evitar o **crescimento excessivo do banco de dados**, automaticamente descartando informações se a diretiva **AutoPrune** (recurso pool) estiver definida para **yes** (padrão) ou se houver rotinas regulares de **prune**. Vamos analisar agora essas outras retenções.

Diretivas File Retention e Job Retention

As duas diretrizes são normalmente⁵⁹ definidas no recurso **Client** do bacula-dir.conf do cliente, permitindo que o administrador defina valores diferentes por cliente do Bacula:

```
Client {  
  Name = someclient-fd  
  Password = "someclientpassword"  
  Address = 192.168.0.15  
  FDPort = 9102  
  Catalog = MyCatalog  
  AutoPrune = yes  
  File Retention = 60 days  
  Job Retention = 6 months  
}
```

- **File Retention.** A tabela de nome **"File"** do banco de dados do Bacula contém um índice de nomes de arquivos copiados, caminho completo e metadados. Isso é o que permite ao administrador listá-los e restaurar um único arquivo de um job de backup inteiro. Sem essa informação só é possível restaurar todo o trabalho a partir de uma interface Bacula.

- **Job Retention.** A informação do job permite que ele seja restaurado pelo Bacula. Sem essa informação, só é possível a restauração através do bextract, ou se o bscan for utilizado no volume para restaurar as informações do catálogo.

Conclusão

Se **AutoPrune = yes** ou um comando **prune (do bconsole)** for agendado para execução regularmente (ex.: **job do tipo admin**), todas as informações de **file** e **job** que ficam além da respectiva retenção são descartadas, apesar de você ainda poder restaurar as informações do **catálogo** sem necessariamente ter uma instalação do Bacula. Neste caso, utilize o bextract, bscan e bls, ferramentas de recuperação de desastres mais lentas e mais complexas, não destinadas ao uso prioritário.

Para novos usuários do Bacula é recomendado aumentar esses valores para números maiores que o tempo de retenção do volume (**Volume Retention**). O Bacula não irá preservar índices de volumes que já foram **reciclados** e isso não terá qualquer inconveniente maior além de seu

catálogo crescer um pouco mais (para uma carga de trabalho média). Se existir, no entanto, uma quantidade excessiva de arquivos de backup dos clientes (muitos milhões de arquivos), esse aumento pode ser melhor. Por exemplo:

```
Client {
  Name = someclient-fd
  ...
  AutoPrune = yes
  File Retention = 1 year
  Job Retention = 1 year
}
```

7.11 Compressão dos dados do backup

A compactação via software do Bacula é mais comumente usada para backups gravados em disco, isso porque a maioria das unidades de fita já tem a compressão via hardware habilitada (feita pelo próprio drive⁶⁰, mais eficiente). Ter as duas compressões habilitadas seria redundante, sendo que, no caso específico do Bacula Community, pode haver alguma vantagem, pois a compressão do Bacula acontece no cliente, o que afetaria os dados trafegados, economizando alguma capacidade de rede.⁶¹

Para usar a compactação **via software**, certifique-se de que o **cliente** do Bacula foi compilado com as bibliotecas de desenvolvimento necessárias respectivas **gz** e/ou **lzo** (veja o capítulo de instalação anterior)⁶². Em seguida, você deve incluir a diretiva a seguir no sub-recurso **Options, Include**, em todos os recursos **FileSet** em **bacula-dir.conf** nos quais deseja copiar os arquivos compactados. No caso do **GZIP**, é possível ajustar também o nível de compressão:

```
FileSet {
  Name = "apache_app"
  Include {
    Options {
      signature = MD5
      Compression = GZIP1
    }
    File = /var
    File = /etc
  }
  ...
}
```

GZIP tem de **1 a 9** níveis de compressão que podem ser escolhidos, onde **9** comprime mais, ao custo de mais gasto com processamento (não é aconselhável usar mais do que o **nível 6**, que é o padrão por omissão, pois o custo-benefício é baixo). Também é possível usar o algoritmo **LZO**, mais leve e relativamente mais rápido que o **GZIP1**. A seguir, veja um comparativo das performances dos algoritmos suportados⁶³:

	Original		
GZIP CPU usage	100%	100MB	
GZIP2	100%	100MB	
GZIP3	100%	100MB	
GZIP4	100%	100MB	
GZIP5	99%	100MB	
GZIP6ZIP	98%	100MB	
GZIP7	97%	100MB	
GZIP8	97%	100MB	

High CPU usage	97.58%		
Very low CPU usage	148.11 MB		
(Example data set: 437 MB; yours will give different results.)			

Tabela 7.1. Algoritmos de compressão – resultados de teste de desempenho do Bacula.

Observação 1: se você tiver vários armazenamentos diferentes, você ainda poderá desativar a compactação (por dispositivo) usando a diretiva **AllowCompression = no**, no recurso storage do bacula-dir.conf.

Observação 2: para o mesmo **FileSet** você pode ter mais de um sub-recurso “include”. Dessa forma, é possível selecionar algumas pastas ou arquivos que você deseja que o Bacula comprima e outros que não.

7.12 Jobs de cópia e migração

Os jobs de cópia e migração (**copy** e **migration**) do Bacula são ótimos para mover dados de backup de hardware de armazenamento antigo para novo, executar backups **disk-to-disk-to-tapes** (disco para disco para fitas) ou implementar um segundo **armazenamento off-site**.

A cópia e a migração acontecem no nível de granularidade de job, dessa forma o tamanho da fita ou mídia de origem ou destino não é importante. Os jobs a serem copiados de uma **pool (e storage)** Bacula de **origem** devem ser selecionados a partir de alguns critérios definidos e são armazenados em uma **pool (e storage) de destino**.

A **diferença** entre os jobs de cópia e migração é que o Bacula descarta os jobs gravados da pool de origem no caso da **migração**.

Quando o **backup original** de um job copiado é reciclado ou excluído por algum motivo, o Bacula selecionará automaticamente a cópia para fazer uma restauração de um cliente naquele determinado momento, se solicitado.

Os trabalhos de cópia e migração **não requerem** conexão com os **clientes do Bacula**. Isso geralmente significa que eles podem ser executados durante o dia, sem risco de impacto nos demais serviços.

O escopo de seleção dos jobs para a cópia pode ser:

- Um ou vários jobs de backup configurados.
- Um ou vários volumes, pelo nome.
- Jobs de um cliente, pelo nome.
- Jobs realizados que correspondam a uma expressão regular.
- Se um trabalho existe em um determinado volume.
- Volumes com o somatório de ocupação em uma pool inferior ou superior a um dado valor.
- Tamanho do volume.

Em resumo, para executar um **job** de cópia ou migração, você deve definir um recurso job especial em bacula-dir.conf com a diretiva **Type** igual a **Copy** ou **Migrate**; a especificação de uma **pool** de origem; e, pelo menos, mas não menos importante, um dos **SelectionType** (e

muitas vezes **SelectionPattern**) que define os critérios de seleção capazes de encontrar jobs para serem copiados.

Nos recursos **pool** (ainda em bacula-dir.conf), você deve especificar na pool de origem a diretiva **Storage** (a mesma nas quais seus volumes já estão sendo gravados, que também será a origem da cópia), além da diretiva **NextPool**, que deve informar o nome de **outra pool** de destino da cópia. Por fim, no recurso da **pool de destino**, também deve ser especificada uma diretiva **Storage** para um dispositivo de armazenamento diferente, que será o **de destino**.

Agora vamos ver mais informações sobre cada uma dessas diretrizes necessárias. Mais adiante há também um exemplo completo de configuração.

Para o recurso Job

Pool =

Determina onde o algoritmo de seleção de trabalhos para cópia será aplicado. É a pool de origem.

Type = Migrate

Os jobs selecionados são movidos da **pool** de origem para a de destino (os trabalhos originais são descartados). **Ou...**

Type = Copy

Os jobs são copiados para a **pool** de destino. Eles receberão os tempos de retenção e reciclarão o comportamento da nova pool, de igual sorte.

Selection Type =

Determina como os **JobIds** para cópia ou migração serão selecionados. Esses são os valores possíveis:

- **SmallestVolume** – Seleciona todos os jobs do menor volume da pool de origem.
- **OldestVolume** – Seleciona todos os jobs do volume mais antigo.
- **Client** – Se utilizado, requer também uma outra diretiva (**SelectionPattern**), que deve ser uma expressão regular que corresponda aos **nomes dos clientes** para cópia/migração apenas dos jobs dos clientes.
- **Volume** – Se utilizada, esta diretiva também requer a **SelectionPattern**, neste caso uma expressão regular que corresponderá aos **nomes de volume** cujos jobs serão copiados.
- **Job** – Neste caso, a **SelectionPattern** deve corresponder a nomes de jobs.
- **SQLQuery** – Uma **query SQL** para encontrar JobIds diretamente do banco de dados Bacula a serem copiados.
- **PoolOccupancy** – O Bacula **somará** o tamanho de todos os volumes da pool de origem. Se for menor do que as diretivas acessórias (necessárias no próprio recurso da pool) **MigrationLowBytes** ou superior a **MigrationHighBytes**, ele migrará os trabalhos até que a ocupação da pool volte aos limites definidos.
- **PoolUncopiedJobs** – Bastante popular para backups off-site. Ele copiará todos os trabalhos que não tiverem uma cópia na pool de destino.

Selection Pattern =

Esta diretiva não é necessária para **SelectionType OldestVolume** nem **SmallestVolume**,

mas indispensável para **Client**, **Volume** e **Job**. O valor de `selectionPattern` deve ser uma expressão regular que irá encontrar os respectivos nomes de diretiva.

Para o recurso pool de origem

Migration Time =

Um período de tempo no qual, quando acabar, os jobs de backup podem ser migrados se um job de migração for executado.

Migration High Bytes =

Tamanho máximo da ocupação da pool.

Migration Low Bytes =

Tamanho mínimo da ocupação da pool.

Next Pool =

Especifica o que é a pool de destino da cópia (obrigatório). No entanto, também é possível substituir essa especificação de **NextPool** no **recurso Schedule**, permitindo que você execute jobs de cópia ou migração para pools de destino diferentes de acordo com o tempo.

Storage =

Neste caso, o nome do **storage de origem**⁶⁴ deve ser especificado, o mesmo que provavelmente já vem sendo utilizado para gravar os backups desta pool. De qualquer sorte, este é o melhor lugar para associar as pools aos seus storages, independentemente de estar configurando um job de cópia.

Para o recurso pool de destino

Storage =

O nome do **storage de destino** deve ser especificado, onde os jobs de cópia ou migração serão armazenados.

7.12.1 Um exemplo de configuração de job de cópia

Recursos pool (bacula-dir.conf)

```
# Pool de Origem
```

```
Pool {
    Name = Monthly-Disk
    Pool Type = backup
    Storage = File1          # storage de origem
    Next Pool = Copy         # nome da pool de destino
    ...
}
```

```
# Pool de destino
```

```
Pool {
    Name = Copy
    Pool Type = backup
    Storage = File2          # storage de destino
    ...
}
```

Job Resource (bacula-dir.conf):

```

Job {
    Name = "CopyJobs"
    JobDefs = DefaultJob
    # Type = Migration
    # Pool = Migration-Disk
    Selection Type = PoolUncopiedJobs
    # Schedule = MigrationCopySchedule
}

```

Aplique as alterações na configuração e **execute** (comando **run**) o job criado como teste. Neste exemplo, a primeira cópia pode levar muito tempo, uma vez que irá selecionar todos os jobs nunca copiados da pool de origem (**SelectionPattern** = **PoolUncopiedJobs**). Portanto, planeje o seu início cuidadosamente, pois essa primeira carga de trabalho pode ser grande. Você também deve querer criar e especificar para este job um **schedule** diferente de seu backup regular, como no exemplo.

7.13 Etiquetagem automática de volumes

A etiquetagem automática é o recurso do Bacula que cria volumes conforme necessário de acordo com alguns parâmetros. Ela deve ser usada para backups baseados em disco, mas também pode ser usada para unidades de fita manuais. As bibliotecas de fitas não precisam disso, porque os nomes de volume devem corresponder ao nome em etiquetas de código de barras⁶⁵.

O **Bacula** só criará novos volumes se não houver volumes com status append (graváveis) ou não puder limpar (**purge**) os antigos com retenção expirada. Você pode ver no log de backup as mensagens de criação de volume quando isso acontece.

O **LabelFormat** (formato do rótulo) é definido no **recurso Pool** (**bacula-dir.conf**) e você pode desativar ou ativar a funcionalidade no **recurso Device** (**bacula-sd.conf**) da seguinte maneira:

bacula-dir.conf

```

Pool {
    Name = Diaria-Disco
    Pool Type = backup
    ...
    LabelFormat = "${Pool}-${NumVols}"
    Maximum Volume Bytes = 50G
    Maximum Volumes = 50
}

```

bacula-sd.conf

```

Device {
    Name = File
    Media Type = File
    Archive Device = /home/bacula/backups
    LabelMedia = yes
    ...
}

```

Se você não especificar variáveis como eu fiz (ex.: **LabelFormat** = **Vol**), o **Bacula** vai anexar uma sequência de números a fim de gerar nomes de volume exclusivos (por exemplo:

Vol0001, Vol0002), que correspondem a MediaID. Isso poderia ser confuso quando há várias pools com o label automático, uma vez que a ordem de numeração das fitas seria distribuída de maneira aleatória entre as diferentes pools.

Pessoalmente, prefiro utilizar `Label Format = "${Pool}-${NumVols}"` porque é simples, gerará nomes de volume únicos e todos os volumes na mesma pool seguirão uma ordem numérica na criação.

Não há necessidade de usar variáveis de data aqui, já que o Bacula já armazena no catálogo a data de **label** (criação), que pode ser buscada usando o comando **llist media** do bconsole. Se ainda assim você quiser, no entanto, o exemplo pode ser seguido para acrescentar a data no formato ISO:

```
"${Pool}-${NumVols}-${Year}-${Month: p/2/0/r}-${Day: p/2/0/r}"
```

No final deste livro há um anexo com as variáveis do Bacula e algumas opções.

A diretiva **Maximum Volume Bytes** (que define o tamanho máximo dos volumes) deve ter um **valor razoável** e é indispensável para backups em disco (caso contrário, um volume apenas iria consumir o disco inteiro e seria inteiramente descartado quando fosse reciclado). Os volumes são alocados dinamicamente, portanto nem todos os volumes ocuparão **50G (exemplo anterior)**, dependendo de como outras diretivas limitadoras do uso de volumes estiverem configuradas.

Você deve distribuir seu espaço de armazenamento em disco para um número de volumes que sejam suficientes para cumprir minimamente sua estratégia de backup. Por exemplo: para GFS clássico com 1 ano de retenção máxima você deve ter, pelo menos: **4 volumes diários; 4 semanais e 12 mensais**, dado o fato de você provavelmente usar um volume diferente para cada um desses dias. Neste caso, eu dividiria meu disco por **20**.

Entretanto, não há nenhum problema em ter mais volumes do que a sua estratégia sugere, uma vez que o Bacula suporta backups multivolume automaticamente. Basta não usar um **Maximum Volume Bytes** muito baixo, uma vez que você teria centenas de volumes e tornaria difícil **visualizar a listagem destes** na sua interface de administração.

A outra diretiva que pode afetar a rotulagem automática de volume é a **Maximum Volumes = 50**. Isso é necessário se você armazenar backups na mesma partição de um sistema operacional ou simplesmente não pode deixar o Bacula consumir completamente a partição de storage em disco.

A propósito: se o seu Bacula não tiver espaço em disco suficiente para criar novos volumes, ele irá bloquear backups e enviar mensagens de e-mail pedindo mais espaço ou reciclar manualmente **(purge) volumes de backup usados**.

7.14 Deduplicação em nível de arquivos⁶⁶

A principal razão para usar a **deduplicação do Bacula Community** é se você tiver exatamente os mesmos arquivos entre várias máquinas diferentes. Este pode ser o caso se seus sistemas operacionais são exatamente os mesmos e atualizados para a última versão ou se você tem

várias máquinas com a mesma versão de aplicativos.

Essa deduplicação funciona da seguinte maneira: você deve executar um backup marco (especial) que tenha um nível específico dentro do Bacula: **Base Job** (sempre uma espécie de **Full**), que deve ser executado também em uma pool diferente. Todas as cópias de segurança do cliente do Bacula que estão configuradas para comparar o seu conteúdo com as do trabalho base **não repetem a cópia** dos mesmos arquivos que já foram copiados pelo trabalho de **Base**.

Se a deduplicação estiver configurada corretamente, você poderá ver no registro do log dos jobs a proporção de arquivos que não foram backupeados porque foram copiados no **Base Job** correspondente.

Em teoria, você poderia configurar um backup de **Base** de uma máquina e comparar apenas seus backups futuros com sua **Base**. O efeito prático disso é que, mesmo que você envie um trabalho de backup **Full**, o Bacula somente copiará os arquivos que foram alterados desde a última terminação do trabalho nível **Base**. Seria basicamente o mesmo comportamento de executar um backup **Full** e, em seguida, um **diferencial/incremental**. Portanto, não há uma vantagem clara de usar a deduplicação para uma única máquina específica.

7.14.1 Configurando a deduplicação do Bacula

Adicione o novo recurso **Job de base especial** no **bacula-dir.conf**. É aconselhável executar isso em uma pool diferente que você deve criar, e também é aconselhável executá-lo em uma **schedule** diferente, uma vez que é provável que você vai executá-lo em uma frequência mais baixa:

```
Job {  
    Name = BaseJob  
    Level= Base  
    Pool = "Base-Pool" # é preciso criar.  
    ...  
}
```

Ainda no **bacula-dir.conf**, adicione as seguintes diretivas aos trabalhos que podem ter os arquivos similares ao **Job de Base** e que por isso serão deduplicados:

```
Job {  
    Name = BackupHeitor  
    Base = BackupHeitor, BaseJob  
    Accurate = yes  
    Schedule = base_schedule  
    FileSet = debian_7_set  
    ...  
}
```

A diretiva **Base** diz ao Bacula o universo de tarefas de backup regulares e a base que serão comparados e não repetir a cópia de arquivos semelhantes. No exemplo correto, o trabalho do **BackupHeitor** está se comparando com o trabalho chamado **BaseJob**. A diretiva **Accurate = yes** também é obrigatória.

Não deixe o **bacula-dir.conf** ainda. Você também deve fazer algumas alterações em seu FileSet do backup regular original:

```
FileSet {
  Name = debian_7_set
  Include {
    Options {
      BaseJob = pmugcs5
      Accurate = mcs5
      Verify = pin5
    }
    File = /etc
    File = /var
    File = /opt
  }
  ...
}
```

Cada uma dessas opções necessárias estabelece comportamentos diferentes para a forma como o Bacula procura e compara arquivos entre jobs de backup **Base** e **regulares**. Eles são os mesmos suportados para o **Verify Job** (job de verificação) do Bacula.

Adicione ao menos uma **nova pool e schedule** para as tarefas de backup base. Considere usar um valor significativo para **Volume Retention** que não seja menor do que os backups regulares; caso contrário, você poderia **perder a capacidade de executar** a restauração completa de alguns **jobs Full** se o trabalho **base** a que eles se referem **já estiver reciclado**.

```
Pool {
  Name = Base-Pool
  Pool Type = Backup
  Volume Use Duration = 18 hours
  Volume Retention = 364 days
  ...
}

Schedule {
  Name = base_schedule
  Run = Differential Pool=Base-Pool 1st sunday at 12:00
}
```

Realize uma tarefa **Base** e, em seguida, a tarefa de backup regular (ex.: **Full**) agora deduplicada. Você pode notar que informações semelhantes aparecerão no resumo do log do job:

```
...
Rate: 2425.4 KB/s
Software Compression: 39.7 %
Base files/Used files: 39336/39114 (99.44%)
VSS: yes
Encryption: no
...
```

7.15 Criptografia do tráfego das comunicações do Bacula (TLS)

TLS⁶⁷ é um código de criptografia de rede interno para fornecer transporte de rede seguro semelhante ao oferecido pelo stunnel ou ssh. Os dados gravados em volumes pelo daemon de armazenamento **não são criptografados** por esse código (**apenas a comunicação**). Para criptografia de dados, consulte o tópico “Criptografia dos dados de backup”, mais adiante.

Basicamente, o administrador pode configurar a criptografia para todas as comunicações de serviço do Bacula (**bconsole → Director, Director → Client, Director → Storage, Client → Storage**). No entanto, o tráfego de **client para o storage** contém as informações mais críticas (dados reais de backup e restauração). Alguns administradores do Bacula só ativam a criptografia desse tráfego.

Os recursos suportados neste código incluem:

- Negociação TLS cliente/servidor requerida.
- Conexões com validação de certificado de servidor e cliente TLSv1.
- Suporte de sigilo direto via Diffie-Hellman de codificação efêmera⁶⁸.

7.16 Informações de diretivas e configuração TLS⁶⁹

Diretrizes de configuração adicionais são suportadas por todos os daemons (Director, daemon de arquivo e daemon de armazenamento), bem como por vários programas de console diferentes. Essas novas diretivas são definidas da seguinte forma:

TLS Enable = yes|no

Ativar suporte TLS. Se o TLS não estiver ativado, nenhuma das outras diretivas TLS terá efeito. Em outras palavras, mesmo se você definir TLS Enable = yes, você precisa ter o TLS habilitado ou o TLS não será usado.

TLS Require = yes|no

Exige conexões TLS. Esta diretiva é ignorada a menos que o TLS Enable esteja definido como yes. Se o TLS não for necessário e o TLS estiver ativado, o Bacula se conectará a outros daemons com ou sem TLS, dependendo do que o outro daemon solicite. Se o TLS estiver **ativado e requerido**, o Bacula recusará qualquer conexão que não use o TLS.

TLS Certificate = Filename

O caminho completo e nome de arquivo de um certificado TLS codificado PEM. Ele pode ser usado como um certificado de cliente ou servidor. PEM significa Privacy Enhanced Mail, mas neste contexto refere-se a como os certificados são codificados. Ele é usado porque os arquivos PEM são codificados em base 64 e, portanto, baseados em texto ASCII em vez de binário. Eles também podem conter informações criptografadas.

TLS Key = Filename

O caminho completo e o nome do arquivo de uma chave privada TLS codificada PEM. Deve corresponder ao certificado TLS.

TLS Verify Peer = yes|no

Verifica os pares de certificado. Instrui o servidor a solicitar e verificar o certificado x509 do cliente. Qualquer certificado de cliente assinado por uma CA conhecida será aceito a menos que a diretiva de configuração **TLSAllowedCN** seja usada, caso em que o certificado do cliente deve corresponder ao IP ou nome da máquina (qualificado) especificado. Esta diretiva é válida apenas para um servidor e não em um contexto de cliente.

TLS Allowed CN = ip_fqdn

Atributo de nome comum dos certificados dos pares permitidos. Se esta diretiva for especificada, todos os certificados de servidor serão comparados com esses valores. Isso pode ser usado para garantir que somente o Director aprovado pela CA possa se conectar. Esta diretiva pode ser especificada mais de uma vez. **Você pode usar o endereço IP do seu computador ou o nome completo (qualificado) como o valor CN.**

TLS CA Certificate File = Filename

O caminho completo e o nome do arquivo que especificam um certificado PEM codificado TLS CA. Vários certificados são permitidos no arquivo. Um do TLS CA Certificate File ou TLS CA Certificate Dir são necessários em um contexto de servidor se TLS Verify Peer (citado antes) também for especificado e são sempre necessários em um contexto de cliente.

TLS CA Certificate Dir = Directory

Caminho completo para o diretório de certificados CA do TLS. Na implementação atual, os certificados devem ser armazenados como PEM codificado com hashes compatíveis com OpenSSL, que é hash do nome do assunto e uma extensão de “.0”. Um TLS CA Certificate File ou TLS CA Certificate Dir é necessário em um contexto de servidor se o TLS Verify Peer também for especificado e é sempre necessário em um contexto de cliente.

TLS DH File = Directory

Caminho para o arquivo PEM de parâmetro Diffie-Hellman codificado. Se esta diretiva for especificada, a troca de chaves DH será usada para a codificação efêmera, permitindo o sigilo das comunicações. A troca de chaves DH acrescenta um nível adicional de segurança porque a chave usada para criptografia/descriptografia pelo servidor e pelo cliente é calculada em cada extremidade e, portanto, nunca é passada pela rede se a troca de chaves Diffie-Hellman for usada. Mesmo se a troca de chaves DH não for usada, a chave de criptografia/descriptografia é sempre passada criptografada. Esta diretiva só é válida dentro de um contexto de servidor.⁷⁰

7.16.1 Informações do certificado

Você pode criar um certificado autoassinado para uso do Bacula com TLS que permitirá que você o faça funcionar, mas não permitirá a validação do certificado através de uma autoridade certificadora. Observe, no entanto, que certificados autoassinados funcionarão para o fim de criptografia das conexões. Por exemplo, no caso de o **Director** fazer uma conexão com um **Client**, o daemon de arquivo pode ser configurado para permitir certificados autoassinados, mas o certificado usado pelo Director deve ser assinado por um certificado que é explicitamente confiável no arquivo de configuração do cliente. **Este método é o descrito neste livro.**

Ele serve para evitar ataques **man in the middle** utilizando softwares como o **Ettercap**⁷¹. Essencialmente, se o Director não verificar se está falando com um daemon confiável, ele pode ser enganado e conversar com uma terceira parte maliciosa que está retransmitindo e capturando todo o tráfego, apresentando seus próprios certificados ao **Director** e ao **File Daemon**. A única maneira de evitar isso é usando certificados de uma autoridade

certificadora, de modo que o “homem do meio” seja incapaz de realizar **spoofing** da conexão.

7.16.2 Configurando a criptografia das comunicações

Construindo o Bacula com suporte a criptografia

Para compilar o Bacula com suporte a criptografia (padrão), certifique-se de possuir as bibliotecas **OpenSSL** e cabeçalhos instalados. Ao configurar o Bacula, use:

```
./configure --with-openssl ...
```

Configuração do OpenSSL

Você pode usar uma única máquina (por exemplo: a do Bacula Director) para gerar certificados para ele e todos os outros computadores clientes em que você deseja configurar a criptografia. Edite o arquivo de configuração do OpenSSL:

```
vi /etc/ssl/openssl.cnf
```

Descomentar (ativar) esta linha:

```
nsCertType = server
```

Gerando os certificados autoassinados

Crie e/ou vá para um diretório onde você deseja armazenar seus certificados e mantê-los organizados:

```
cd /etc/bacula/
```

Gere um **par de chaves** para cada máquina **cliente** cujas comunicações com o servidor de backup você deseja criptografar, incluindo o **File Daemon** que é executado na máquina do **Director** (ex.: a do próprio Bacula). Neste caso, o arquivo **.pem** que contém o certificado e a chave é válido por dez anos (**3.650 dias**):

```
openssl req -new -x509 -nodes -out bacula.pem -keyout bacula.pem -days 3650
openssl req -new -x509 -nodes -out example-fd.pem -keyout example-fd.pem -
days 3650
```

Preencha as perguntas solicitadas. Você pode simplesmente responder a cada uma delas digitando um retorno ou pode inserir seus próprios dados, EXCETO para o COMPLETE NAME (nome completo). Deve ser o IP de seu cliente ou o FQDN. Nós vamos precisar desses valores mais tarde para autorizar as conexões (TLSAllowedCN).

Copie o respectivo certificado privado e o par de chaves para cada uma de suas próprias máquinas dos clientes:

```
scp ./example-fd.pem ./bacula.pem client_ip://etc/bacula/
```

Configuração dos daemons do Bacula

Este exemplo de configuração destina-se a criptografar todas as comunicações dos daemons e, portanto, você pode querer alterá-las adequadamente. O **Storage Daemon** está na mesma máquina do **Director** e os itens associam a relação das **diretivas** ao longo dos diferentes arquivos de configuração (por exemplo: grupo (I) para fazer o **bconsole** → **Director** de conexão criptografada; (II) para **Director** → **Storage**; (III) para **Director** → **File Daemon**; e (IV) onde a criptografia deve ser útil: **File Daemon** → **Storage**).

- bconsole.conf:

```
Director {
```

```

Name = backup1-dir
address = 192.168.0.50
...
# (I)
TLS Enable = yes
TLS Require = yes
TLS Certificate = /etc/bacula/bacula.pem
TLS CA Certificate File = /etc/bacula/bacula.pem
TLS key = /etc/bacula/bacula.pem
}

```

•bacula-dir.conf:

```

Director {
    Name = bacula-dir
    ...
    # (I)
    TLS Enable = yes
    TLS Require = yes
    TLS Verify Peer = no
    TLS Allowed CN = "192.168.0.50" # Ou nome qualificado
    TLS CA Certificate File = /etc/bacula/bacula.pem
    TLS Certificate = /etc/bacula/bacula.pem
    TLS Key = /etc/bacula/bacula.pem
}

Storage {
    Name = File
    Address = 192.168.0.50
    ...
    # (II)
    TLS Require = yes
    TLS CA Certificate File = /etc/bacula/bacula.pem
    # Este é um certificado de cliente, usado pelo Director para se
    conectar ao daemon de armazenamento
    TLS Certificate = /etc/bacula/bacula.pem
    TLS Key = /etc/bacula/bacula.pem
}

```

```

Client {
    Name = backup1-fd
    Address = 192.168.0.50
    ...
    # (III)
    TLS Enable = yes
    TLS Require = yes
    TLS CA Certificate File = /etc/bacula/bacula.pem
    TLS Certificate = /etc/bacula/bacula.pem
    TLS Key = /etc/bacula/bacula.pem
}

```

```

Client {
    Name = debian2-fd # Cliente Remoto
    Address = 192.168.0.121
    ...
    # (III)
    TLS Enable = yes

```

```

TLS Require = yes
TLS CA Certificate File = /etc/bacula/debian2-fd.pem
TLS Certificate = /etc/bacula/debian2-fd.pem
TLS Key = /etc/bacula/debian2-fd.pem
}

```

•bacula-fd.conf:

```

Director {
    Name = bacula-dir
    ...
    # (III)
    TLS Enable = yes
    TLS Require = yes
    TLS Verify Peer = yes
    # Permitir apenas que este Director conecte
    TLS Allowed CN = "192.168.0.50"
    TLS CA Certificate File = /etc/bacula/bacula.pem
    # Este é um certificado de servidor. Ele é usado quando "diretores" se
    conectam para verificar a autenticidade deste File Daemon
    TLS Certificate = /etc/bacula/bacula.pem
    TLS Key = /etc/bacula/bacula.pem
}

```

```

FileDaemon {
    Name = debian2-fd
    ...
    # (IV)
    # Você precisa dessas entradas TLS para que o FD e o SD possam se
    comunicar.
    TLS Enable = yes
    TLS Require = yes
    # Certificado da máquina do Storage Daemon.
    TLS CA Certificate File = /etc/bacula/bacula.pem
    TLS Certificate = /etc/bacula/bacula.pem
    TLS Key = /etc/bacula/bacula.pem
}

```

•Segundo bacula-fd.conf (ex.: cliente remoto):

```

Director {
    Name = bacula-dir
    ...
    # (III)
    TLS Enable = yes
    TLS Require = yes
    TLS Verify Peer = no
    # Permitir apenas que este Diretor conecte.
    TLS Allowed CN = "192.168.0.121"
    TLS CA Certificate File = /etc/bacula/debian2-fd.pem
    TLS Certificate = /etc/bacula/debian2-fd.pem
    TLS Key = /etc/bacula/debian2-fd.pem
}

```

```

FileDaemon {
    Name = debian2-fd
    ...
    # (IV)

```



```

# Você precisa dessas entradas TLS para que o FD e o SD possam se
comunicar.
TLS Enable = yes
TLS Require = yes
# Certificado da máquina do Storage Daemon
TLS CA Certificate File = /etc/bacula/bacula.pem
TLS Certificate = /etc/bacula/bacula.pem
TLS Key = /etc/bacula/bacula.pem
}

•bacula-sd.conf:
Storage {
    Name = bacula-sd
    ...
    # (IV)
    # Essas opções de configuração TLS são usadas para conexões recebidas
    do FileDaemon.
    TLS Enable = yes
    TLS Require = yes
    # O certificado de par não é necessário/solicitado - a validade dos
    pares é verificada pelo cookie de conexão de armazenamento fornecido ao
    File Daemon pelo Director.
    TLS Verify Peer = no
    TLS CA Certificate File = /etc/bacula/bacula.pem
    # Este é um certificado de máquina. Ele é usado conectando daemons de
    arquivos para verificar a autenticidade desse Storage Daemon.
    TLS Certificate = /etc/bacula/bacula.pem
    TLS Key = /etc/bacula/bacula.pem
}

Director {
    Name = bacula-dir
    ...
    # (I)
    TLS Enable = yes
    TLS Require = yes
    # Exija que o diretor de conexão forneça um certificado com o CN
    correspondente.
    TLS Verify Peer = yes
    TLS Allowed CN = "192.168.0.50"
    TLS CA Certificate File = /etc/bacula/bacula.pem
    TLS Certificate = /etc/bacula/bacula.pem
    TLS Key = /etc/bacula/bacula.pem
}

```

7.17 Criptografia dos dados de backup

O Bacula permite a criptografia e assinatura dos arquivos de backup copiados dentro do **File Daemon** (cliente) antes de enviar dados para o **Storage Daemon**. Após a restauração, as assinaturas de arquivos são validadas e quaisquer correspondências incorretas são relatadas.⁷² Em nenhum momento o Director ou Storage Daemon têm acesso a conteúdo de arquivo não criptografado, mas aos metadados como nomes e permissões de caminho de arquivo. Isso é prático, já que o administrador de backup pode selecionar quais arquivos serão restaurados sem ter acesso ao conteúdo do arquivo.

Se sua biblioteca de fitas ou armazenamento em disco, no entanto, já possuir recursos de criptografia de hardware, poderá ser mais eficiente usar a criptografia via hardware em vez da criptografia do Bacula, uma vez que ela usa mais recursos de processamento da máquina cliente.

A criptografia e a assinatura do Bacula são implementadas usando chaves privadas RSA associadas a certificados públicos x509 autoassinados. Isso também é conhecido como PKI ou infraestrutura de chave pública.

Cada File Daemon deve receber seu próprio **par de chaves** particulares/públicas. Além deste par de chaves (que é usado para operações regulares de backup e restauração), qualquer número de “**master keys**” (chaves mestras) pode ser gerado e especificado para fins de contingência (recomendado). Estas chaves podem ser usadas para descriptografar quaisquer backups **se a chave File Daemon original for perdida**. Apenas o **certificado público** da chave mestra deve estar disponível para o **File Daemon**. Sob nenhuma circunstância a **chave mestra privada** deve ser compartilhada ou armazenada na máquina do cliente.

As **chaves mestras** podem ser usadas para restaurar dados de qualquer cliente, por isso elas devem ser copiadas para um local seguro, como um DVD ou um pendrive, e colocadas em um cofre ou caixa de depósito de segurança. As chaves mestras **nunca devem ser mantidas** na mesma máquina que o **Storage Daemon** ou o **Director**, pois qualquer acesso eventual não autorizado compromete a confidencialidade de todos os backups.

Embora menos crítica do que as chaves mestras, **as chaves dos File Daemon** são também candidatas preferenciais para backup off-site. Também tenha uma cópia delas em local seguro.

Observação: se você perder suas chaves de criptografia, os backups serão irrecuperáveis. Guarde sempre uma cópia das suas chaves principais num local seguro e diferente do local do centro de dados.

O algoritmo básico utilizado para cada sessão de backup (job) é:

- O File Daemon (FD) gera uma chave de sessão.
- O FD criptografa essa chave de sessão via PKE para todos os destinatários (File Daemon e de todas as chaves mestras).
- O FD usa essa chave de sessão para executar criptografia simétrica nos dados.

7.17.1 Construindo o Bacula com suporte a criptografia

Para criar o Bacula com suporte a criptografia (padrão), certifique-se de possuir as bibliotecas OpenSSL e os cabeçalhos instalados. Ao configurar o Bacula, use:

```
./configure --with-openssl ...
```

7.17.2 Detalhes técnicos da criptografia

A implementação usa AES-CBC 128bit, com chaves de sessão simétricas criptografadas RSA. A chave RSA é fornecida pelo usuário. Se você estiver executando OpenSSL 0.9.8 ou posterior, o hash de arquivo assinado usa SHA-256. Caso contrário, SHA-1 é usado.

As definições de configuração do utilizador final para os algoritmos não estão atualmente expostas – apenas os algoritmos listados antes são utilizados. No entanto, os dados gravados no volume suportam algoritmos arbitrários simétricos, assimétricos e de “digest” para extensibilidade futura, e a implementação de backend suporta atualmente:

Criptografia Simétrica:

- 128, 192, and 256-bit AES-CBC
- Blowfish-CBC

Criptografia assimétrica (usada para criptografar chaves de sessão simétricas):

- RSA

Algoritmos de Digest:

- MD5
- SHA1
- SHA256
- SHA512

Os vários algoritmos são expostos através de uma API OpenSSL totalmente reutilizável (ou seja, é possível cair em um novo backend de criptografia). O formato de volume é DER-codificado ASN.1, modelado após a sintaxe de mensagem criptográfica RFC 3852.

Gerando chaves de criptografia privadas e públicas

Você pode usar uma única máquina (por exemplo: a do Bacula Director) para gerar certificados para ele e todos os outros computadores clientes em que você deseja configurar a criptografia.

Crie e/ou mude para o diretório onde você deseja armazenar seus certificados e mantê-los organizados:

```
cd /etc/bacula/
```

Gere um par de chaves mestras (privada + pública) com:

```
openssl genrsa -out master.key 2048
openssl req -new -key master.key -x509 -out master.cert
cat master.key master.cert > master.pem
```

Mova o arquivo **master.pem** para uma mídia externa segura. Vamos usá-lo mais tarde apenas na eventualidade de recuperação de desastres (**descriptografia de emergência com uma chave mestra**).

Gere um par de chaves do File Daemon para cada backup que você deseja criptografar:

```
openssl req -new -x509 -nodes -out example-fd.pem -keyout example-fd.pem
```

O comando anterior fará **um número de perguntas**. Você pode simplesmente responder a cada uma delas digitando com a tecla “Enter” ou, se desejar, pode inserir seus próprios dados.

Copie a **chave pública mestra** e o respectivo **par de chaves do cliente** para cada uma das suas próprias máquinas:

```
scp ./master.cert client_ip://etc/bacula/
scp ./example-fd.pem client_ip://etc/bacula/
```

Configure os clientes do Bacula para criptografar os dados

```
•bacula-fd.conf
FileDaemon {
    Name = example-fd
    FDport = 9102
    WorkingDirectory = /var/bacula/working
    Pid Directory = /var/run
    Maximum Concurrent Jobs = 20
    PKI Signatures = Yes
    PKI Encryption = Yes
    PKI Keypair = "/etc/bacula/fd-example.pem"
    PKI Master Key = "/etc/bacula/master.cert"
}
```

Descodificação de emergência com uma chave mestra

É preferível manter uma cópia segura, não criptografada, do próprio par de chaves de criptografia do cliente. Se você perder o par de chaves do cliente, a recuperação com o par de chaves mestras é possível.

Você precisa:

- Recuperar sua chave mestra de uma mídia de reserva de reposição armazenada em local seguro.
- Copiar o arquivo que contém o **par de chaves mestras** para a máquina cliente.
- Definir a diretiva **PKIKeypair** em seu arquivo **bacula-fd.conf**, recurso **FileDaemon**:

```
PKI Keypair = master.keypair
```
- Iniciar a restauração. O par de chaves mestras será usado para descriptografar os dados do arquivo **sobre qualquer outra especificação PKI**.

7.18 Pool Scratch

A pool **Scratch** é a única com um nome reservado codificado. É uma pool com capacidade especial de distribuição de volume. Cada vez que outras pools do Bacula estão fora de volumes graváveis, ele procura na pool Scratch fitas utilizáveis, geralmente novas. Isso é muito desejado para bibliotecas de fitas, pois pode ser muito demorado distribuir manualmente fitas recentemente criadas (rotuladas) em diferentes pools que você pode ter; então você pode apenas selecionar a pool **Scratch** como aquela para receber novas fitas ao usar o comando de rótulo de códigos de barras (**label barcodes**).

Para backups de disco, a pool Scratch não é normalmente usada, pois é mais automático deixar o Bacula **autoetiquetar** volumes (padrão).

Observação 1: é possível especificar nomes diferentes e talvez mais de uma pool de Scratch. Realize isso usando a diretiva **"Scratch Pool"** na pool que você deseja receber volumes a partir do zero. Por exemplo:

```
Pool {
    Name = Daily-Tape
    Scratch Pool = NewTapes
    ...
}
```

Observação 2: também é possível, após todo o ciclo de vida do volume (escrita, retenção,

reciclagem), configurá-los para irem de uma pool de backup de volta para a **Scratch**. Isso pode ser desejado se o tamanho dos backups flutuar muito. Isso é feito usando a diretiva **“RecyclePool”** na pool regular cujos volumes você deseja enviar de volta para a **Scratch**. Por exemplo:

```
Pool {  
    Name = Daily-Tape  
    Recycle Pool = Scratch  
    ...  
}
```

7.19 Clientes de backup em redes diferentes

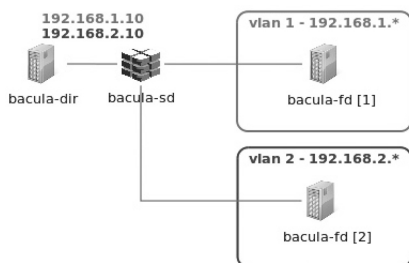


Figura 7.5. Fazendo backup de máquinas de redes diferentes.

Desde a versão 7.0 do **Bacula** é possível especificar endereço de daemon de armazenamento diferente por cliente, e agora é muito mais fácil configurar o sistema de backup para máquinas ao longo de diferentes segmentos de rede.

Na figura, o **bacula-sd** escuta conexões em dois endereços diferentes, e cada cliente tem sua própria especificação de conexão. Para cada configuração de recurso **Client** em **bacula-dir.conf** você pode especificar um endereço diferente:

```
Client {  
    Name = client1-fd  
    Address = 192.168.1.50  
    FD Storage Address =192.168.1.10  
    ...  
}  
  
Client {  
    Name = client2-fd  
    Address = 192.168.2.75  
    FD Storage Address =192.168.2.10  
    ...  
}
```

Essa especificação prevalece sobre o valor da diretiva **Storage Address** especificado no **bacula-dir.conf**, **Storage Resource**.

⁴³ Também é necessário usar o comando **delete** do **bconsole** para remover informações das pools que não estão mais definidas no **bacula-dir.conf**.

⁴⁴ Veja mais detalhes sobre a compressão no próximo capítulo.

⁴⁵ Também é possível usar a diretiva do último dia ao especificar a programação de backup; no entanto, você pode terminar com grandes backups full começando entre dias úteis. Para obter mais opções de especificação de

horário, consulte http://www.bacula.org/7.4.x-manuals/en/main/Configuring_Director.html#SECTION00175000000000000000.

46 Retenções de job e file (banco de dados) podem ser especificadas alternativamente no recurso de pool, e os valores serão aplicados a todos os jobs de backup que gravam lá.

47 Para ambientes grandes, pode ser útil definir AutoPrune igual a “No”, mas criando um novo admin job específico com o comando de prune, para apagar informações em um horário agendado diferente do tempo de backup, evitando concorrência.

48 Alguns usuários também criam uma pool para cada tipo de backup (por exemplo: bancos de dados, sistemas de arquivos etc.), mas adicionam uma complexidade desnecessária à operação do Bacula.

49 É improvável, mas possível, ter mais de um Director capaz de conectar, fazer backup e restaurar o mesmo Bacula File Daemon. Neste caso, haverá mais de um **Director Resource** no arquivo bacula-fd.conf (não considerando “Monitor = yes”, que é outro tipo de configuração).

50 Ajuste o URL do wget para um novo código de versão do Bacula, no meu site ou em <http://bacula.org>.

51 As barras invertidas no comando configure são apenas para quebra de linha cosmética (é um único comando).

52 Ou pelo site do projeto Bacula.org: <http://blog.bacula.org/binary-download-center/>.

53 A diretiva **Monitor = yes** só é importante se o administrador quiser usar o Bacula Tray Monitor, que normalmente não é o caso e é opcional: é seguro remover ou comentar esta sub-rotina para evitar futuras confusões.

54 A instalação do cliente do Windows já solicita essas informações no assistente de instalação. Este não é um déjà-vu. =)

55 Se você tiver qualquer problema ao iniciar o Bacula Client no Windows, é possível iniciá-lo usando a linha de comando (cmd) para verificar as mensagens de erro. Por exemplo: “C:\Program Files\Bacula\bacula-fd.exe” -c “C:\Program Files\Bacula\bacula-fd.conf”.

56 Algumas interfaces gráficas web também têm seu próprio recurso ou gerenciador de ACL, como Community Webacula e Enterprise BWEB.

57 O acento não é suportado na configuração do Bacula.

58 Eu escolhi a primeira sexta-feira do mês para executar backups mensais para evitar confusão, uma vez que os meses podem ter quantidade diferente de sextas-feiras.

59 Também é possível, alternadamente, mover essas configurações para o recurso pool.

60 É por isso que as fitas têm dois valores de capacidade – por exemplo: 20/40 GB. 20 é a capacidade nativa e 40 se a unidade de fita tiver compactação de hardware ativada.

61 O Bacula Enterprise já compacta o tráfego independentemente dos dados.

62 Os pacotes do Bacula já construídos normalmente suportam ambos.

63 Slides do curso Bacula Enterprise Administrator II. Bacula Systems Copyright – todos os direitos reservados. Disponível em: <http://www.baculasystems.com/bacula-administrator-ii-training-course>.

64 Esse storage já deve existir no bacula-dir.conf na forma de um recurso.

65 O comando do bconsole para criar volume nas fitas do robô de fitas é: **label barcodes**.

66 Essa deduplicação é totalmente diferente da deduplicação do Endpoint Global, do Bacula Enterprise.

67 Transport Layer Security.

68 Cifras anônimas Diffie-Hellman não são suportadas por este código. O uso de cifras anônimas DH aumenta a complexidade do código e coloca confiança explícita sobre a implementação bidirecional CRAM-MD5. O CRAM-MD5 está sujeito a ataques de texto simples conhecidos e deve ser considerado consideravelmente menos seguro do que a autenticação baseada em certificado PKI.

69 Fonte: http://www.bacula.com.br/manual/Bacula_TLS_Communications.html.

70 Para gerar o arquivo de parâmetro, você pode usar openssl: **openssl dhparam -out dh1024.pem -5 1024**

71 <http://ettercap.github.io/ettercap/>.

72 Fonte: http://www.bacula.org/7.4.x-manuals/en/main/Data_Encryption.html#SECTION00401000000000000000.

8

Dispositivos de Armazenamento em Disco

8.1 Limitando o tamanho do volume de disco

A principal diferença entre as fitas e os discos é que você pode ter, neste último, **vários volumes** de backup armazenados no mesmo disco ao limitar seu tamanho na diretiva `MaximumVolumeBytes` (`bacula-dir.conf`, recurso `Pool`). Isso geralmente é obrigatório porque **a reciclagem de volume é uma ação atômica (tudo ou nada)**, então ter apenas um volume em seu armazenamento em disco (sem qualquer outro limitador de uso) levaria eventualmente a crescer e preencher o disco inteiro. Ao reciclar, você teria que descartar todos os seus backups anteriores, uma vez que você não teria mais espaço nesse armazenamento.

Não há número mágico limitando o tamanho do volume do disco, uma vez que é apenas o máximo e eles são alocados dinamicamente (por exemplo: **se você executar um trabalho de 1 GB em um volume com capacidade de no máximo 10 GB, ele só ocupará 1 GB**). Portanto, você deve ter um volume mínimo suficiente para ser capaz de ajustar sua estratégia de rotação (ex.: **1 ano GFS com backups diários, semanais e mensais requer pelo menos 20 volumes, dado que você só tem backups monovolumes**). Se você estiver usando apenas um único armazenamento para backups, normalmente recomendo dividir seu tamanho total disponível por **25 ou 30**, e este será o seu `MaximumVolumeBytes`.

```
Pool {
  Name = Diaria-Disco
  Pool Type = Backup
  Recycle = yes
  AutoPrune = yes
  Volume Use Duration = 18 hours
  Volume Retention = 6 days
  Maximum Volume Bytes = 50G
  Maximum Volumes = 100
  Label Format = "${Pool}-${NumVols}"
}
```

Você pode ter notado a partir do exemplo anterior que você ainda pode limitar a quantidade máxima de volumes por pool (`Maximum Volumes`). Isso é mais útil se você tiver um sistema operacional ou aplicativos em execução no mesmo dispositivo de armazenamento de backup e não desejar arriscar o preenchimento inteiro do disco com backups e quiser limitar de alguma forma.

8.1.1 Configurando o dispositivo de disco no `bacula-sd.conf`

A nova instalação do Bacula traz dois conjuntos de dispositivos para armazenamento de

backup: **FileChgr1** e **FileChgr2**. Essa configuração funciona de forma semelhante a uma biblioteca de fitas, mas para discos. Você pode ter vários dispositivos de gravação para um único ou vários discos.

```
Autochanger {
  Name = FileChgr1
  Device = FileChgr1-Dev1, FileChgr1-Dev2
  Changer Command = /dev/null
  Changer Device = /dev/null
}
```

```
Device {
  Name = FileChgr1-Dev1
  Media Type = File1
  Archive Device = /backup
  ...
}
```

```
Device {
  Name = FileChgr1-Dev2
  Media Type = File1
  Archive Device = /backup
  ...
}
```

No exemplo anterior você tem dois dispositivos de escrita virtual para um mesmo disco, então você eventualmente acaba escrevendo em dois volumes diferentes, em diferentes setores de disco ao mesmo tempo. Se você estiver fazendo trabalhos em multiplexação (jobs simultâneos), isso leva a um melhor desempenho.

Para outros discos, storages ou partições, é necessário criar (ou utilizar) outro Autochanger, com outro valor de Archive Device ou Media Type.

O **Bacula SD** possui algumas diretivas de montagem automática de disco para o sistema operacional e para dispositivos removíveis, a exemplo de storages USB externos, mas para **discos internos** é mais fácil usar uma partição já montada permanentemente (como o exemplo: **/backup**), normalmente editando o arquivo **fstab** do Linux.

Você também vai usar **fstab** se deseja utilizar **NAS** para escrever backups do Bacula. O mais recomendado aqui é habilitar capacidades **iSCSI** no lado do servidor que roda o Storage Daemon; configurar o **fstab** para montá-lo sempre em tempo de boot; e definir, no **bacula-sd.conf**, a diretiva **ArchiveDevice** com valor de acordo com o diretório selecionado do ponto de montagem. O uso de outros protocolos não é recomendado para escrita ou leitura de backups por conta de limitações de desempenho (nfs, cifs, smb etc.).

Limitar o máximo de trabalhos simultâneos para cada dispositivo (**Maximum Concurrent Jobs** = 5) levará o Bacula a realizar automaticamente um **balanceamento de carga entre os dispositivos de gravação virtuais**.

É sempre bom lembrar que se você adicionar um novo Autochanger (grupo de dispositivos) em **bacula-sd.conf**, você também deve **amarrá-lo** ao seu Director (**bacula-dir.conf**) para daí poder configurar jobs que escrevam nele. Por exemplo:

```
Storage {
  Name = DiskStorage
```



```

Address = 192.168.0.10
SDPort = 9103
Password = "44wYG+7FPXVTQ/sTj/RiiAtpmTaN7yN4v"
Device = FileChgr1
Media Type = File1
Maximum Concurrent Jobs = 10
Autochanger = yes
}

```

O nome do storage (criação livre) deve corresponder depois ao job, JobDefs, pool ou schedule. Você pode selecionar o armazenamento para backups em qualquer um desses recursos, mas acho que especificar na pool é geralmente melhor, uma vez que funciona melhor tendo apenas volumes de um único armazenamento (matriz rígida que prevalece sobre storage definida em job e JobDefs).

O **Device** e **Media Type** devem corresponder exatamente aos valores indicados para a configuração de **Autochanger** em **bacula-sd.conf**; caso contrário, seu Director **não se conectará** ao Storage Daemon (isso pode ser testado com um comando **status storage**, do **bconsole**).

8.1.2 VPS e outros hosts remotos como armazenamento de backup (ex.: Amazon AWS)

O primeiro instinto de uso do Bacula para armazenar backups no Amazon Storage Services (S3) é usar o **s3fs** para montá-lo no local em uma máquina que executa um Bacula Storage Daemon. Isso pode levar a um desempenho muito ruim, uma vez que usa um protocolo de camada 7 e não é viável para uma quantidade significativa de dados⁷³.

O método recomendado aqui é alugar uma máquina virtual gerenciável no seu provedor de nuvem (por exemplo: Amazon EC2), instalar um Daemon de Armazenamento do Bacula nele e configurar o seu Director Bacula e clientes locais para escrever backups lá. É muito mais seguro e funcional, uma vez que o Bacula tem características favoráveis como: controle de largura de banda; cópia e migração etc. **Ignore a subseção a seguir para ver as etapas para implantar um segundo ou remoto Bacula Storage Daemon.**

Montando o Amazon S3 Bucket localmente

Para este guia, tente sempre usar as versões mais recentes do software baixado. Remova pacotes antigos se instalados e resolva dependências. Como root:

•CentOS

```

yum remove fuse fuse-s3fs && yum install gcc libstdc++-devel gcc-c++
automake curl-devel libxml2-devel openssl-devel mailcap libfuse-dev
libssl-dev

```

•Debian

```

apt-get remove fuse && sudo apt-get install build-essential libcurl4-
openssl-dev automake autotools-dev libfuse-dev libssl-dev libxml2-dev
mime-support

```

É hora de buscar o código do software **fuse**, extrair para **/usr/src** e compilá-lo. Para ambas as distribuições, basta executar os comandos:

```
wget -T 3 -qO- \
https://github.com/libfuse/libfuse/releases/download/fuse-2.9.7/
fuse-2.9.7.tar.gz | tar -xzvf - -C /usr/src
```

```
cd /usr/src/fuse-2.9.7
export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
./configure --prefix=/usr/local
make -j8 && make install
```

Alguns módulos de configuração:

```
echo /usr/local/lib >> /etc/ld.so.conf
echo include ld.so.conf.d/*.conf >> /etc/ld.so.conf
```

Carregue o módulo do **fuse**:

```
ldconfig
modprobe fuse
```

Vamos fazer o mesmo para **s3fs**. Primeiro faça o download, extraia, configure e faça (**por favor, use a versão mais recente disponível para s3fs**):

```
wget -T 3 -qO- https://github.com/s3fs-fuse/s3fs-fuse/archive/v1.80.tar.gz \
| tar -xzvf - -C /usr/src
```

```
cd /usr/src/s3fs-fuse-1.80
export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
./autogen.sh && ./configure --prefix=/usr/local
make -j8 && make install
```

No console AWS, procure a sua chave secreta (**secret_access_key**) e seu **ID**, uma vez que serão necessários para a autenticação do software:

```
echo AWS_ACCESS_KEY_ID:AWS_SECRET_ACCESS_KEY > ~/.passwd-s3fs
chmod 600 ~/.passwd-s3fs
```

Configure um dispositivo (Autochanger) de gravação em disco no **bacula-sd.conf** para usar um ponto de montagem desejado (por exemplo):

```
Archive Device = /mnt/bucket
```

Crie as pastas necessárias para o **bucket** e o cache local:

```
mkdir /mnt/cache
mkdir /mnt/bucket
chmod 777 /tmp/cache /mnt/bucket
```

O comando de montagem deve ser algo como a seguir. Substitua conforme necessário:

```
s3fs -o url=http://s3-us-west-2.amazonaws.com baculaheitor -o use_cache=/tmp/
cache /mnt/s3fs
```

O valor **baculaheitor** é o nome do bucket; **url** é o endereço do seu bucket, que pode ser buscado nas propriedades do console AWS.

É uma **boa ideia** testar e depurar adicionando as opções **-d -f** para o comando de montagem antes de prosseguir.

Ao final, configure o valor da diretiva **Device > Archive Device** de **bacula-sd.conf** para corresponder ao ponto de montagem do bucket.

Para desmontar o bucket, eventualmente:

```
fusermount -u /mnt/s3fs
```

8.1.3 Implantando um segundo e/ou Bacula Storage Daemon remoto

Na máquina Linux remota, instale pacotes **bacula-sd** do repositório ou construa o **Storage Daemon** a partir do código-fonte. **Sua versão precisa corresponder à versão do Director**, e a compilação deve incluir as mesmas dependências, a escolha das bibliotecas do mesmo banco de dados usado para o catálogo do Bacula. Para **construir apenas o Storage Daemon**, acrescente as opções a seguir ao **./configure** completo mostrado no capítulo de instalação inicial do servidor completo do Bacula:

```
./configure --enable-build-dird=no --enable-build-stored=yes --with-mysql (or postgresql etc.)
```

Prossiga com o **make**, **make install**, **make install-autostart** (etc.) e edite o arquivo **bacula-sd.conf** do SD remoto:

```
Director {
    Name = debian-dir # modifique para o nome real do seu director
    Password = "storage_password" # apenas copie esse valor randomicamente
    # gerado para amarração futura ao director
}

Autochanger {
    Name = FileChgr1
    Device = FileChgr1-Dev1, FileChgr1-Dev2
    Changer Command = /dev/null
    Changer Device = /dev/null
}

Device {
    Name = FileChgr1-Dev1
    Media Type = File1
    Archive Device = /mnt/backup # ajuste para o diretório local ou ponto de
    # montagem em que deseja gravar os volumes de backup
    LabelMedia = yes
    Random Access = yes
    AutomaticMount = yes
    RemovableMedia = no
    AlwaysOpen = no
    Maximum Concurrent Jobs = 5
}

Device {
    Name = FileChgr1-Dev2
    Media Type = File1
    Archive Device = /mnt/backup # ajuste para o diretório local ou ponto de
    # montagem em que deseja gravar os volumes de backup
    LabelMedia = yes
    Random Access = Yes
    AutomaticMount = yes
    RemovableMedia = no
    AlwaysOpen = no
    Maximum Concurrent Jobs = 5
}
```

Certifique-se de que seu ponto de montagem no disco existe, está configurado

corretamente e que o usuário bacula tem privilégios de escrita:

```
chown bacula /mnt/backup
```

Volte para seu servidor de backup (Director). Modifique o arquivo **bacula-dir.conf**:

```
Storage {
    Name = Off-Site-Storage
    Address = 172.245.35.120 # IP da máquina do Storage Daemon
    SDPort = 9103
    Password = "storage_password" # deve ser igual à senha copiada do bacula-
sd.conf remoto.
    Device = FileChgrl # deve ser igual ao nome do Autochanger correspondente em
bacula-sd.conf.
    Media Type = File1 # Também precisa ser igual ao definido no Autochanger do
bacula-sd.conf remoto.
    Maximum Concurrent Jobs = 10
}
```

Salve e reinicie o Director e o Storage Daemon na máquina remota. Teste a conexão com o comando do bconsole **status storage**.

Uma maneira simples de definir jobs para gravar no **segundo Storage Daemon** é criar uma nova pool com a definição de armazenamento. Para enviar ou agendar backups para ele:

```
Pool {
    Name = Off-site-pool
    Pool Type = Backup
    Recycle = yes
    AutoPrune = yes
    ...
    Label Format = "${Pool}-${NumVols}"
    Storage = Off-Site-Storage
}
```

Exemplo de schedule (agendamento) simples:

```
Schedule {
    Name = "Off-Site"
    Run = Full Pool=Off-Site-Storage 1st sun at 23:05
}
```

Todos os jobs de backup submetidos à pool off-site serão salvos no Storage Daemon remoto. Você também pode testá-lo executando um job de trabalho ad hoc (avulso – comando run).

Observação: consulte o item “Jobs de cópia e migração” neste livro se você desejar copiar backups feitos localmente para esse segundo armazenamento.

Backup Bacula em Fitas Magnéticas

Pode-se dizer que as fitas ainda são o tipo de mídia mais comum para o armazenamento de backup de empresas de médio e grande porte. Elas possuem um custo razoável por gigabyte, grande velocidade de gravação, grande capacidade de armazenamento e durabilidade. As fitas são dispositivos de escrita sequencial, por isso é quase impossível descartar ou substituir partes de informações já gravadas. Quando recicladas, elas são geralmente “apagadas” rapidamente (com uma marca final da mídia gravada no início dela), que informa ao sistema operacional que a mídia está em branco (dados reais ainda estão lá, mas não podem mais ser facilmente recuperados).

O hardware de fita magnética mais recomendado são as bibliotecas de fitas, embora você ainda possa encontrar drives manuais em algumas empresas. Quanto à visão do sistema operacional, a única diferença entre eles é que as bibliotecas de fitas também têm pelo menos um dispositivo de troca automática de fita, também conhecido como picker ou medium-changer (medium-x), e talvez mais de uma unidade de fita.

A maioria das unidades de fita vem com especificação de capacidade, por exemplo: **1.5/3.0 TB**. A primeira capacidade, também chamada de nativa, é a esperada se você por algum motivo desativar capacidades de compressão feita pelo drive de fitas. A compressão via hardware é mais eficiente do que um software, ainda que o Bacula suporte o backup via software com diferentes algoritmos. Você pode verificar se a compactação da unidade de fita está habilitada com um comando Linux (por exemplo): `tapeinfo -f /dev/sg3`

9.1 Dispositivos de robôs de fitas

Normalmente eles são conectados fisicamente via SCSI na máquina que será o servidor, com ao menos um cabo para cada drive, sendo possível até dois para fins de balanceamento de carga entre ambos. O controlador SCSI precisa ser compatível com o seu dispositivo de biblioteca de fita, por isso é importante verificar a matriz de compatibilidade ao planejar a instalação.

Eles também têm uma porta ethernet regular exclusiva para o acesso ao console web. Pode ser bastante útil para solucionar problemas, para eventualmente mover as fitas manualmente ou simplesmente para reiniciar o robô remotamente.

No Linux, a maioria dos dispositivos de biblioteca de fitas pode trabalhar com unidades que vêm com o sistema operacional e podem funcionar como dispositivos genéricos **SCSI**. É muito recomendável ter os pacotes **sg_utils** e **lsscsi** instalados, ou o equivalente à sua distribuição, para configurar e usar o seu trocador automático de fitas (autochanger).

O comando do shell **lsscsi -g** é ótimo para verificar os caminhos dinâmicos do dispositivo para seus dispositivos e se eles são reconhecidos pelo seu sistema operacional:

```
lsscsi -g
[0:0:0:0] disk ATA VBOX HARDDISK 1.0 /dev/sda /dev/sg0
[1:0:0:0] cd/dvd VBOX CD-ROM 1.0 /dev/sr0 /dev/sg1
[2:0:0:0] cd/dvd VBOX CD-ROM 1.0 /dev/sr1 /dev/sg2
[3:0:0:0] mediumx STK L700 0105 /dev/sch0 /dev/sg11
[3:0:1:0] tape IBM ULT3580-TD5 0105 /dev/st0 /dev/sg3
[3:0:2:0] tape IBM ULT3580-TD5 0105 /dev/st1 /dev/sg4
```

Eu posso ver o dispositivo trocador de fitas (**mediumx = medium changer**) e as duas unidades de fita (**linhas em negrito**), então tudo deve estar pronto para a configuração do Bacula, mas ainda não estaremos utilizando esses caminhos na configuração do Bacula Storage Daemon, uma vez que eles podem mudar sua ordem toda vez que a máquina for reinicializada.

A solução para isso é usar links permanentes (by-id) que a maioria das distribuições Linux cria automaticamente para dispositivos de fita:

```
ls -la /dev/tape/by-id/
total 0
drwxr-xr-x 2 root root 400 Sep 22 14:17 .
drwxr-xr-x 4 root root 80 Sep 22 14:17 ..
lrwxrwxrwx 1 root root 9 Sep 22 14:17 scsi-350223344ab000100 -> ../../st0
lrwxrwxrwx 1 root root 10 Sep 22 14:17 scsi-350223344ab000100-nst -> ../../nst0
lrwxrwxrwx 1 root root 9 Sep 22 14:17 scsi-350223344ab000200 -> ../../st1
lrwxrwxrwx 1 root root 10 Sep 22 14:17 scsi-350223344ab000200-nst -> ../../nst1
lrwxrwxrwx 1 root root 10 Sep 22 14:17 scsi-SSTK_L700_XYZZY_A -> ../../sg11
```

Há mais coisas que você precisa considerar aqui:

- 1.O Bacula lida melhor com as chamadas de unidades de fita na forma não rebobinável. Por exemplo: `/dev/tape/by-id/scsi-350223344ab000100`
- 2.A ordem das unidades (índice) na configuração do daemon Bacula Storage deve corresponder à ordem física real de instalação na sua biblioteca de fitas. Vamos tentar descobrir essa ordem.

Uma maneira de buscar a ordem das unidades de biblioteca de fita é carregar qualquer fita em uma das unidades que você tem. Você pode usar o comando **mtx** para fazer isso:

```
mtx -f /dev/sg11 status
Storage Changer /dev/sg11:2 Drives, 8 Slots
Data Transfer Element 0:Empty
Data Transfer Element 1:Empty
Storage Element 1:Full :VolumeTag = E01001L4
Storage Element 2:Full :VolumeTag=E01002L4
Storage Element 3:Full :VolumeTag=E01003L4
Storage Element 4:Full :VolumeTag=E01004L4
Storage Element 5:Full :VolumeTag=E01005L4
Storage Element 6:Full :VolumeTag=E01006L4
Storage Element 7:Full :VolumeTag=E01007L4
Storage Element 8:Full :VolumeTag=E01008L4
```

Ele irá listar todas as unidades de uma fita que você tem em sua biblioteca de fitas e que seu

sistema irá enxergar.

```
mtx -f /dev/sg11 load 1 0
```

Vai carregar uma fita que você tem no slot hipotético (1) na primeira unidade (0). Você pode fazer outro **mtx ... status** para certificar-se de que a fita foi inserida.

```
mtx -f /dev/sg11 status
```

```
Storage Changer /dev/sg11:2 Drives, 8 Slots
```

```
Data Transfer Element 0:Full (Storage Element 1 Loaded):VolumeTag = E01001L4
```

```
Data Transfer Element 1:Empty
```

Agora use o seguinte comando para cada uma das suas unidades de fita, para verificar qual está on-line/retorna saída/tem um código de densidade diferente (depende da sua distribuição Linux):

```
mt -f /dev/nst0 status
```

```
mt: /dev/nst1: rmtopen failed: No medium found
```

```
mt -f /dev/nst1 status
```

```
drive type = 114
```

```
drive status = 1174405120
```

```
sense key error = 0
```

```
residue count = 0
```

```
file number = 0
```

```
block number = 0
```

Você pode ver que realmente **/dev/nst1** é a primeira unidade (drive 0) de sua biblioteca de fitas, então você deve ter isso em mente ao configurar o arquivo **bacula-sd.conf**. A seguir estão as alterações mínimas para esse exemplo específico⁷⁴:

```
Autochanger {
```

```
    Name = STK_L700
```

```
    ...
```

```
    Changer Device = /dev/tape/by-id/scsi-SSTK_L700_XYZZY_A
```

```
    Device = Drive-1
```

```
    Device = Drive-2
```

```
}
```

```
Drive {
```

```
    Name = Drive-1
```

```
    ...
```

```
    Drive Index = 0
```

```
    Media Type = LTO
```

```
    Archive Device = /dev/tape/by-id/scsi-350223344ab000200-nst
```

```
}
```

```
Drive {
```

```
    Name = Drive-1
```

```
    ...
```

```
    Drive Index = 1
```

```
    Media Type = LTO
```

```
    Archive Device = /dev/tape/by-id/scsi-350223344ab000100-nst
```

```
}
```

Depois de configurar o seu **bacula-sd.conf** corretamente, você deve amarrar seu novo dispositivo criado ao seu Director, adicionando um novo recurso Storage:

```
Storage {
```

```
    Name = Name = STK_L700 # Nome Fantasia
```

```

Address = 192.168.0.1 # Endereço da máquina do SD
SDPort = 9103
Password = "XgCxmjGoekEai6YVybJopSq+Lad9RQk0N"
Device = STK_L700 # Precisa ser igual em bacula-sd.conf
Media Type = LTO
Autochanger = yes
}

```

Reinicie o Director e o Storage Daemon. Você deve ser capaz de buscar informações de biblioteca de fitas no **bconsole** com o comando `status storage`.

Agora deve ser hora de rotular suas fitas (logicamente). Nesse momento você já deve ter colado os rótulos de código de barras fisicamente sobre elas. Basta usar o seguinte comando e o Bacula criará seus volumes lógicos em todas as fitas que ainda não as tenham. No **bconsole**:

```
label barcodes
```

Normalmente, você selecionará a pool **Scratch** para ser a hospedeira dos novos volumes a serem criados, pois trata-se de uma maneira mais automática de manipulação de fitas. O Bacula vai buscar fitas em branco na pool **Scratch** cada vez que houver demanda para escrita de jobs em outras pools. Não há necessidade em distribuir, manualmente, as novas fitas entre as diferentes outras pools de backup.

9.2 Outras operações de fitas

9.2.1 O comando mt

Normalmente, você não precisa fazer as operações diárias de Bacula. Mas se você já usou fitas em outra instalação de software de backup, talvez seja necessário apagá-las. Esses são alguns exemplos de uso:

```

mt -f /dev/nst0 rewind # rebobina fita
mt -f /dev/nst0 weof # apaga rápido
mt -f /dev/nst0 erase # apaga realmente e devagar
mt -f /dev/nst0 eject # ejeta a fita do drive

```

Você não deve apagar fitas escritas pelo Bacula se você pretende continuar fazendo seu backup através dele. Várias informações importantes são dadas pelo rótulo da fita e trata-se de um processo contraproducente apagá-lo toda vez que desejar sobrescrevê-la. O comando **purge** do **bconsole** é a maneira mais elegante de forçar o Bacula a descartar um conteúdo de fita e torná-lo pronto para ser substituído.

9.3 Manipulação de robôs de fitas

Situação 1: cada vez que você inserir novas fitas você **precisa** fazer o Bacula estar ciente das mudanças; ou talvez você simplesmente queira verificar quais fitas estão no robô de fitas.

- Você deve usar o comando **update slots** do **bconsole**:

```
*update slots
```

```
Automatically selected Catalog: MyCatalog
```

```
Using Catalog "MyCatalog"
```

```
The defined Storage resources are:
```

```
1: File
```



```
2: STK_L700
```

```
Select Storage resource (1-2): 2
```

Escolha o seu dispositivo de biblioteca de fitas. Se solicitado para qual dos drives, você pode escolher qualquer um (essa pergunta é irrelevante para este comando). Esta será a saída:

```
Enter autochanger drive[0]:
```

```
Connecting to Storage daemon ultrium-lto2-Changer at 192.168.0.1:9103 ...
3306 Issuing autochanger "slots" command. Device "STK_L700" has 8 slots.
3306 Issuing autochanger "list" command.
Volume "E01001L4" not found in catalog. Slot=1 InChanger set to zero.
Volume "E01002L4" not found in catalog. Slot=2 InChanger set to zero.
Volume "E01003L4" not found in catalog. Slot=3 InChanger set to zero.
Volume "E01004L4" not found in catalog. Slot=4 InChanger set to zero.
Volume "E01005L4" not found in catalog. Slot=5 InChanger set to zero.
Volume "E01006L4" not found in catalog. Slot=6 InChanger set to zero.
Volume "E01007L4" not found in catalog. Slot=7 InChanger set to zero.
Volume "E01008L4" not found in catalog. Slot=8 InChanger set to zero.
```

Se não for encontrado no catálogo (**not found in catalog**), significa que sua instalação do Bacula não tem volumes criados nessas fitas ainda. Você provavelmente deseja continuar com o comando `label barcodes` e tentar criar volumes nelas, a menos que elas já tenham sido escritas por outra instalação do Bacula ou outro software de backup cujos dados você ainda precisa preservar.

Situação 2: “eu sei que não é necessário para a operação diária, mas como posso mover mídia da biblioteca de fitas com o Bacula?”

• **Você pode usar o comando `mount/umount` para carregar/descarregar fitas no `bconsole`.** Você será solicitado a informar a **origem/destino** e o **drive/slot**.

```
*mount
```

```
Automatically selected Catalog: catalog Using Catalog "catalog"
```

```
The defined Storage resources are:
```

```
1: sata-changer
```

```
2: STK_L700
```

```
Select Storage resource (1-12): 2
```

```
Enter autochanger drive[0]: 0
```

```
Enter autochanger slot: 1
```

```
3307 Issuing autochanger "unload slot 8, drive 0" command.
```

```
3304 Issuing autochanger "load slot 1, drive 0" command. 3305
```

```
Autochanger "load slot 1, drive 0", status is OK. 3001 Mounted Volume:
BACU10
```

```
3001 Device "CHGDRV1" (/dev/nst1) is already mounted with Volume
"BACU10"
```

No exemplo eu pedi para o Bacula carregar a fita que estava no slot 1 do robô no primeiro drive (0). A saída me mostra que já havia uma fita no drive (a do slot 8), então ele a descarrega e somente depois carrega a fita do slot 1.

Situação 3: como eu sei que fitas estão nas unidades ou o que o Bacula está fazendo com elas?

• Use o comando **status storage** do `bconsole`:

```
Device "Drive-1" (/dev/nst0) is mounted with:
```

```
Volume: BC0032L4
```

```
Pool: Tape-Daily
Media type: LTO4
Slot 1 is loaded in drive 0.
Total Bytes Read=64,512 Blocks Read=1 Bytes/ block=64,512
Positioned at File=0 Block=0
Device "Drive-2" (/dev/nst1) is not open.
Drive 1 is not loaded.
```

Além das informações da fita (se existe uma fita na unidade, se existem jobs escrevendo nela etc.), você pode verificar a lista de unidades que pertence à biblioteca de fitas (**Drive-1** e **Drive-2**).

No primeiro drive temos uma fita do Bacula etiquetada **BC0032L4**, e no segundo drive não temos nenhuma fita carregada.

Durante as operações usuais do Bacula, o ideal é que você não precise fazer qualquer intervenção manual de carregar ou descarregar fitas, mas deixá-lo lidar com elas automaticamente.

9.3.1 O mtx-changer

Este é um script Shell que pertence ao código do Bacula e é responsável pela mudança de fitas. Ele usa o programa **mtx** para executar essas operações.

O script **mtx-changer** tem um arquivo de configuração denominado **mtx-changer.conf** (fica no diretório de scripts do Bacula) que possui alguns valores que, em raras ocasiões, podem ser usados para melhorar a compatibilidade da biblioteca de fitas.

Como exemplo, as diretrizes **offline_sleep** e **load_sleep** podem ter valores maiores se sua biblioteca de fitas precisar de mais lapso de tempo entre os comandos, evitando erros.

A título informativo, se você quiser testar **mtx-changer** ou utilizá-lo para manualmente carregar ou descarregar fitas, esta é a sintaxe:

```
mtx-changer <changer_device> <command> <slot> <tape_drive_device>
<drive_index [o mesmo definido em bacula-sd.conf]>
```

Os valores aceitos em **<command>** são:

unload - unloads tape from drive into selected slot.

load <slot> - loads tape from given slot into the selected drive.

list - list tape-library contents.

loaded - gives the tape information (slot) for a given drive. 0 (zero) means the drive is empty.

slots - provides the number of available slots.

volumes - lists the slots and available volumes

Exemplo de load (carga) simples

```
mtx-changer /dev/sg11 load 11 /dev/nst0 1
```

```
# Carrega a fita do slot 11 no Drive 0 que é o dispositivo /dev/nst0
```

9.4 Escrever jobs de backup em vários drives simultaneamente

Usar várias unidades simultâneas para gravar backup pode soar tentador, mas nem sempre é

desejado, pois você irá espalhar backups através de várias fitas e você pode acabar usando mais mídias do que você normalmente faria, dependendo da sua **política de backup**.

De qualquer forma, existem várias maneiras de gravar em unidades diferentes simultaneamente. No **bacula-dir.conf** há uma diretiva que supostamente faria um balanceamento de carga dos jobs, um-por-um entre os drives disponíveis. No entanto, essa implementação **não é perfeita até o momento e não é recomendada de forma alguma**:

```
PreferMountedVolumes = no
```

Se **PreferMountedVolumes** é igual a **no** (yes é o padrão), o Bacula tentará montar novas fitas para cada tarefa de backup. Isso pode funcionar bem por algum tempo, mas inevitavelmente acabará levando o Bacula Storage Daemon a uma situação de erro.

Outras maneiras mais confiáveis para forçar o Bacula a escrever em fitas diferentes

1. Programar vários jobs de backup **para duas ou mais pools diferentes** ao mesmo tempo e definir esse balanceamento manualmente através da configuração de jobs que serão executados para cada uma das pools. O Bacula será forçado a escrever em uma fita diferente para cada pool de backup.
2. Para cada configuração de drive de fita no **bacula-sd.conf** (recurso Device), defina um limite para jobs simultâneos (por exemplo: **MaximumConcurrentJobs** = 5). Sempre que você tiver mais de cinco trabalhos simultâneos em execução para esse robô de fitas, o Bacula vai pedir uma segunda fita no próximo drive para dividir a escrita.

9.5 Data spooling⁷⁵

Os backups podem ser configurados para serem gravados em um dispositivo de disco antes de serem movidos para a fita, em ato contínuo. Toda vez que o espaço configurado do **spool** está cheio enquanto o backup é feito, os dados são movidos para a fita e o espaço é liberado imediatamente. Isso pode ser útil em algumas situações:

- Para organizar melhor os dados de diferentes jobs de backup nas fitas, pois eles podem ser gravados de maneira mais agrupada nas fitas, exigindo menos leitura de fita ao restaurar um job.
- Para lidar com clientes muito lentos que não forneceria uma taxa de transferência significativa para escrever na fita diretamente, evitando um efeito indesejável de gravação de fita chamado **shoe-shining** (vai e volta/lustrar sapato).
- Se você tiver um disco mais rápido que a fita e com espaço em disco suficiente para guardar em spool uma quantidade significativa de dados (de preferência, igual ao tamanho da fita). Caso contrário, seu backup pode ser mais lento ou não aproveitar ao máximo as vantagens do spool.

Observação: É possível e talvez confiável e mais eficiente (também) utilizar spool para backups gravados em disco se for configurada uma área em SSD para essa finalidade, tornando o tempo de backup dos clientes menor devido à escrita mais rápida.

Essas são as configurações necessárias para o spooling de dados:

Para cada job em **bacula-dir.conf** você pode habilitar ou desabilitar o spooling de dados

conforme desejado (recursos **Job**, **JobDefs** ou **Schedule**):

`SpoolData = yes/no`

No **bacula-sd.conf** desejado (drive de fita device resource), você deve definir o diretório onde os dados serão colocados em spool. Se você não configurar a diretiva seguinte, o Bacula usará seu próprio diretório de trabalho, mas se este não estiver em outra partição pode acabar consumindo todo o seu espaço do sistema operacional:

`Spool Directory = diretório`

Você pode limitar o tamanho máximo para esse spool ainda no recurso de dispositivo bacula-sd.conf (padrão é ilimitado):

`Maximum Spool Size = [tamanho em bytes, gigabytes (GB) etc.]`

Também é possível limitar o tamanho do spool para cada tarefa de backup no mesmo recurso (o padrão é ilimitado). Dessa forma, um job hipotético mais rápido não consumiria todo o tamanho do spool:

`Maximum Job Spool Size = [tamanho em bytes, gigabytes (GB) etc.]`

Observação: nunca exclua o conteúdo do diretório do spool durante os backups!

9.6 Imprimir novas etiquetas de código de barras

As bibliotecas de fitas costumam ter capacidade de leitura de códigos de barras e por isso ser muito mais rápidas. Se você não tiver os rótulos já impressos é muito fácil criá-los e até mesmo personalizá-los. Geralmente há **seis** ou **oito** caracteres no código, e a maioria das bibliotecas permite que o administrador (em seu console da web) escolha seu tamanho (oito é o mais comum).



Figura 9.1. Código de barras.

Fonte: <<https://sep.yimg.com/ay/lenscomputersupplies/lto-ultrium-barcode-labels-7.jpg>>.

Existem várias opções on-line que geram arquivos PDF com as etiquetas. Basta imprimir, cortar e colar nas fitas. Você pode conseguir ainda melhores resultados de impressão em um papel reflexivo, mas os regulares também são bons. Por exemplo: <<http://kelvin.nu/barcode>>.

Basta escolher a tecnologia de fita e a geração (por exemplo: LTO Ultrium Gen 4), um prefixo para suas fitas, que pode ser personalizado, e a numeração desejada.

Depois de carregar a biblioteca com as novas etiquetas, execute um **update slots** (comando **bconsole**) para atualizar as informações no Bacula e ver as novas fitas e um **label barcodes** para criar volumes lógicos no Bacula (com o mesmo nome de rótulo) e torná-los prontos para receber backups adicionais, como visto anteriormente.

9.7 Fitas de limpeza

Fitas de limpeza são mais bem tratadas automaticamente pelo hardware. Isso é feito

reservando um slot de biblioteca de fita para a fita de limpeza (geralmente através de seu console web): ele será inserido no drive sujo toda vez que for necessário (nenhuma intervenção manual necessária). Na verdade, forçar a limpeza do drive quando não é explicitamente exigido por qualquer LED ou outro aviso vai desgastá-lo desnecessariamente, pois as fitas de limpeza utilizam um material químico abrasivo.

Além disso, é altamente recomendável dizer ao Bacula **quais são as fitas de limpeza que estão em seu trocador de fitas antes de tentar rotulá-los!** O Bacula pode reconhecer automaticamente fitas de limpeza com a configuração **"CleaningPrefix = xxx"** das pools, onde xxx deve coincidir com os caracteres na etiqueta de papel. Por exemplo:

```
Pool {
    Name
    ...
    Cleaning Prefix = "CLN"
}
```

9.8 Configuração e automação de drive de fita manual

Dependendo de como você configura o Bacula para acessar suas unidades de fita manuais, pode ser muito frustrante (por exemplo: usando os exemplos comentados presentes nas configurações de fábrica). Você pode precisar montar, desmontar e ejetar fitas manualmente, o que pode ser muito demorado.

A seguinte configuração (**bacula-sd.conf**) usa partes do script **mtx-changer** para automatizar algumas dessas operações, permitindo que o Bacula desmonte e ejele automaticamente fitas ao exigir outra para continuar com operações de backup ou restauração.

```
Device {
    Name = ManualLTO4
    Archive Device = /dev/nst0
    Changer Device = /dev/null
    # Isso é importante. Talvez você precise alterar o diretório de script para
    # o seu caminho real.
    Changer Command = "/etc/bacula/scripts/mtx-changer %o %a %S"
    # Opcional.
    Maximum Changer Wait = 3d
    AutomaticMount = yes
    AlwaysOpen = yes
    Media Type = LTO
    RemovableMedia = yes # Importante.
    Offline On Unmount = Yes
    # Se você usar etiquetas automáticas para tapes.
    Label media = Yes
}
```

Se não for capaz ou não estiver satisfeito com o método anterior, você sempre poderá automatizar algumas das operações com scripts Run After/Before Job:

1.Um script para montar fitas antes da primeira tarefa de backup, ou mesmo no cron, para montar qualquer fita a cada meia hora (por exemplo):

```
exec 6>&1
exec > /etc/bacula/status_scrip.log # cria log do script
```

```
/etc/bacula/bconsole -c /etc/bacula/bconsole.conf <<END mount
storage="Storage_Name"
END
exec 1>&6 6>& -
```

2.Um script para desmontar e ejetar fitas (normalmente APÓS o último job de backup, geralmente o BackupCatalog):

```
exec 6>&1
exec > /etc/bacula/status_script.log # logs the script
/etc/bacula/bconsole -c /etc/bacula/bconsole.conf << END unmount
storage="Storage_Name"
END
exec 1>&6 6>&-
mt -f /dev/nst0 rewind # se necessário substitua nst0
mt -f /dev/nst0 eject
```

⁷⁴ Existe uma configuração mínima de autochanger comentada em cada arquivo bacula-sd.conf fresco. Você também pode querer verificar algumas diretivas de tuning no capítulo correspondente.

⁷⁵ Fonte: <http://www.bacula.org/7.4.x-manuals/en/main/Data_Spooling.html>.

10

Fatores de Performance e Tuning

10.1 Fatores de desempenho do backup

Existem vários recursos computacionais que podem afetar o desempenho do sistema de backup, não apenas o hardware do servidor de backup e o banco de dados do catálogo, mas também máquinas de clientes, interfaces de rede, switches, roteadores, capacidade de gravação ou leitura em disco e armazenamento em fita.

Quanto ao software, sistema de arquivos de armazenamento, tuning do banco de dados do catálogo, configuração do sistema operacional e outras configurações específicas do backup, como tamanho de buffer de tráfego de rede, quantidade de pedaços do volume, multiplexação etc. podem influenciar na velocidade de backup.

Dito isso, a parte mais difícil é identificar em qual componente existe um gargalo, se ele pode ser resolvido ou se o desempenho atual é compatível com as especificações de hardware.

10.2 Identificando gargalos

Normalmente, os recursos de backup exigidos são: velocidade de escrita no armazenamento; largura de banda disponível na rede; e leitura do cliente de backup. A melhor maneira de descobrir com certeza é medindo suas performances.

10.2.1 Benchmarking de discos

Leitura⁷⁶

Um bom software Linux para fazer isso é o **hdparm**. Por exemplo:

```
sudo hdparm -Tt /dev/sda
```

Saída:

```
/dev/sda:
```

```
Timing cached reads: 12960 MB in 2.00 seconds = 6484.97 MB/sec
```

```
Timing buffered disk reads: 398 MB in 3.00 seconds = 132.56 MB/sec
```

Os valores para cada execução podem sofrer uma pequena variação, mas servem como um bom parâmetro para saber a capacidade de leitura na cópia de máquinas com clientes do Bacula.

Escrita

Nesse caso, consiste no uso do antigo comando **dd**.

```
dd if=/dev/zero of=/tmp/output bs=8k count=250k; rm -f /tmp/output
```

Saída:

```
256000+0 records in
256000+0 records out
2097152000 bytes (2,1 GB) copied, 16,5959 s, 126 MB/s
```

No exemplo, escolhi 250k de granularidade para ter dados significativos totais para o teste (2,1 GB), mas um tamanho de teste maior pode ser mais preciso e compatível com sua carga de job real.

Se não existir nenhum sistema de arquivos no disco, pode executar o teste da seguinte forma:

```
dd if=/dev/zero of=/tmp/output bs=8k count=10k; rm -f /tmp/output
```

10.2.2 Benchmarking de fitas

Escrita

O **btape** é um software do Bacula que pode ser usado para testar unidades de fita, fitas e operações. Ele pode ser chamado pelo Shell, mudando o caminho para o dispositivo de unidade de fita ou o caminho para o arquivo bacula-sd.conf:

```
./btape -c /etc/bacula/bacula-sd.conf /dev/nst0
```

Você deve ter conseguido acessar seu prompt:

```
Tape block granularity is 1024 bytes.
btape.c:376 Using device: /dev/nst0
*
```

A velocidade de escrita pode ser testada conforme a seguir:

```
*speed file_size=3 skip_raw
btape.c:1078 Test with zero data and bacula block structure.
btape.c:956 Begin writing 3 files of 3.221 GB with blocks of 129024 bytes.
+++++
btape.c:604 Wrote 1 EOF to "Drive-0" (/dev/nst0)
btape.c:406 Volume bytes=3.221 GB. Write rate = 44.128 MB/s
...
btape.c:383 Total Volume bytes=9.664 GB. Total Write rate = 43.531 MB/s
btape.c:1090 Test with random data, should give the minimum throughput.
btape.c:956 Begin writing 3 files of 3.221 GB with blocks of 129024 bytes.
+++++
btape.c:604 Wrote 1 EOF to "Drive-0" (/dev/nst0)
btape.c:406 Volume bytes=3.221 GB. Write rate = 7.271 MB/s
+++++
...
btape.c:383 Total Volume bytes=9.664 GB. Total Write rate = 7.365 MB/s
```

Os valores usados e outras opções detalhadas para um teste maior estão bem documentados no manual⁷⁷ do Bacula.

10.2.3 Benchmarking das conexões de rede

Este teste é útil para conhecer a real capacidade atual de transferência de rede entre os **File Daemons do Bacula** e a **máquina do Storage Daemon**, que é mais exigente.

O aplicativo **iperf** (Linux e Windows) deve ser instalado em ambas as máquinas para executar o teste. Por exemplo:


```
yum install iperf
```

OU

```
apt-get install iperf
```

Execução do teste

Em uma das máquinas (normalmente aquela que recebe as conexões Storage Daemon), inicie o **iperf** no modo servidor:

```
iperf -s
```

Na outra máquina, inicie o teste de conexão com a anterior. Por exemplo:

```
iperf -c 192.168.0.10
```

A saída deve ser impressa em alguns segundos:

```
-----
Client connecting to localhost, TCP port 5001
TCP window size: 2.50 MByte (default)
-----
[ 3] local 127.0.0.1 port 56882 connected with 192.168.0.10 port 5001
[ ID] Interval Transfer Bandwidth
[ 3] 0.0-10.0 sec 38.0 GBytes 32.7 Gbits/sec
```

10.3 Tuning do servidor do Bacula⁷⁸

Essas são configurações gerais que devem ser sempre testadas e podem não ter qualquer benefício em alguns cenários (é bom saber o que realmente está sendo feito).

10.3.1 Configurações do Bacula

- Fitas: aumente os valores do **Minimum** (por exemplo: 256K) e o **Maximum Block Size** para 512K em **bacula-sd.conf**, recurso **Device** (valores recomendados para LTO4. 1M é muito grande e pode causar desgaste de hardware). É necessário recriar volumes depois de alterar esses valores.
- Fitas: aumente o tamanho máximo do arquivo para 10 GB ou 20 GB (bacula-sd.conf, recurso de dispositivo). Melhores velocidades de escrita, restauração um pouco mais lenta.
- Fitas: ativar o spooling de disco.
- Execute sempre vários jobs simultâneos (diretiva **MaximumConcurrentJobs**). Certifique-se de que valores maiores sejam explícitos em quatro recursos (porque o padrão é apenas um):
 - Director em bacula-dir.conf.
 - Storage em bacula-dir.conf.
 - Storage em bacula-sd.conf.
 - Device em bacula-sd.conf.
- Escreva em vários discos ou armazenamentos simultaneamente.
- Desativar o **Autoprune** de clients e jobs (ao invés, crie um job do tipo Admin para limpar suas informações uma vez por dia em outro job agendado).
- Ative o **Attributespooling** para todos os jobs (normalmente padrão após o Bacula 5.x, recurso JobDefs do bacula-dir.conf).

- Certifique-se de que o banco de dados suporta inserção em lotes – batch insert (normalmente padrão).

- Divida o FileSet ao fazer backup de um número imenso de arquivos (por exemplo: mais de 20 milhões).

- Baixe o nível de compressão GZIP (por exemplo: GZIP1) ou use LZO (diminui o uso da CPU da máquina do File Daemon).

10.3.2 Banco de dados do catálogo (catalog)

PostgreSQL

- Evite a criação de índices adicionais.

- Configurações que geralmente trazem melhoria no desempenho (postgresql.conf):

```
work_mem = 1MB
wal_buffer = 64k
shared_buffers = 1GB # up to 8GB
work_mem = 64MB
effective_cache_size = 2GB
checkpoint_segments = 64
checkpoint_timeout = 20min
checkpoint_completion_target = 0.9
maintenance_work_mem = 256MB
synchronous_commit = on
```

- Realização (periódica) de um vacuumdb na base de dados (Postgresql), pois com o passar do tempo a grande alteração de registros acaba por deixar a inserção no banco mais demorada.⁷⁹

MySQL

Algumas configurações para o MySQL:

```
sort_buffer_size = 2MB
innodb_buffer_pool_size = 128MB
innodb_flush_log_at_trx_commit = 0
innodb_flush_method = O_DIRECT
```

Execute o **mysqltuner** (apt-get install mysqltuner) e verifique alguns itens para melhorar.

10.3.3 Conexões de rede (para o FD e SD)

- Ajuste tamanho máximo do buffer de rede = bytes. O padrão é **32.768** bytes, mas alguns usuários relatam melhor desempenho em redes locais usando 65.536 bytes.

- Adicione mais interfaces de rede e switches mais rápidos.

- Evite firewall e roteadores.

- Use bonding e jumbo frames.

- Customize o kernel (veja <<https://fasterdata.es.net/host-tuning/linux/>>):

```
net.core.wmem_max
net.core.rmem_max
net.ipv4.tcp_rmem
net.ipv4.tcp_wmem
```

10.3.4 Sistema operacional

- Certifique-se de fornecer à máquina do Bacula **8 GB** de RAM.

- Certifique-se de que ele é 64 bits.
- Opções de sysctl do Linux:
vm.dirty_ratio = 2
vm.dirty_background_ratio = 1
vm.swappiness = 10
vm.zone_reclaim_node = 0

10.3.5 Acesso ao disco

- Use RAID com um bom controlador (por exemplo: ARECA).
- Use “deadline disk scheduler”.
- Prefira XFS como sistema de arquivos.

⁷⁶ Disponível em: <<http://askubuntu.com/questions/87035/how-to-check-hard-disk-performance>>.

⁷⁷ Disponível em: <http://www.bacula.org/7.4.x-manuals/en/problems/Testing_Your_Tape_Drive_Wit.html#SECTION00420000000000000000>.

⁷⁸ Muitas dessas dicas foram fornecidas por Kern Sibbald, em palestra sobre o Bacula em 2014 para a Latinoware.

⁷⁹ Referências: <<http://www.postgresql.org/docs/9.0/static/app-vacuumdb.html>> e <<http://www.devmedia.com.br/otimizacao-uma-ferramenta-chamada-vacuum/1710>>.

11

Instalando Interfaces Gráficas

11.1 BAT – Ferramenta de administração Bacula (desktop)

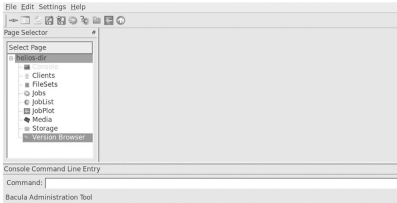


Figura 11.1. tela do BAT.

Esta ferramenta de desktop simples e poderosa destina-se a ser instalada na estação de trabalho do administrador do Bacula e requer conexão com a máquina do Director na porta 9101. Ela não fornece os recursos de edição dos arquivos de configuração do Bacula, mas é ótima para submeter operações do sistema, como submeter jobs de backup, restore etc.

Sua configuração é bastante semelhante ao `bconsole.conf` – por isso, você pode apenas copiar o conteúdo de um `bconsole.conf` funcional para um `bat.conf`:

```
cp /etc/bacula/bconsole.conf /etc/bacula/bat.conf
```

Se você precisar configurar um novo arquivo `bat.conf`, você provavelmente precisará alterar o nome do Director, que deve ser exatamente o mesmo configurado no seu `bacula-dir.conf`; além do endereço IP para a máquina do Director e a senha do console, que está localizada no recurso do Director do `bacula-dir.conf`.

Podemos instalar o **bat** usando pacotes Linux/Windows ou construindo a partir do código-fonte:

Pacotes (Linux)

Ele vem por padrão nos principais repositórios de distribuição. Exemplo do Debian:

```
apt-get install bacula-console-qt
```

Construção a partir do código-fonte (Linux)

- Resolva as dependências. Para Ubuntu ou Debian⁸⁰, este é o caminho a percorrer:

```
apt-get install build-essential pkg-config libqwt5-qt4 libqwt5-qt4-dev
```
- Baixe e extraia o código-fonte do Bacula. No diretório criado, crie binários do Bacula como de costume, mas adicione a opção `enable-bat` (você pode precisar pelo menos do `--enable-client-only` se estiver construindo em sua estação de trabalho):

```
./configure --enable-bat --enable-client-only  
make -j8 && make install
```

Pacote (Windows)

Faça o download dos pacotes compatíveis para Windows do Bacula e selecione a opção “Consoles” na tela “Choose Components”.

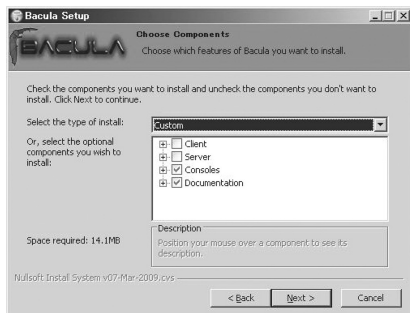


Figura 11.2. Seleção de instalação do BAT Windows.

11.2 Webacula 7.x



Figura 11.3. Logomarca do Webacula.

Esta é uma ótima interface, que também não tem capacidade para alterar os arquivos de configuração do Bacula, mas é ótima para a administração de serviços de backup. O Webacula tem tradução para vários idiomas, é fácil de usar e possui sua própria ACL gráfica, que permite autosserviço (ou autoatendimento) de maneira segura, a exemplo de permitir que usuários não administradores iniciem restauração apenas dos servidores que tenha direitos definidos.

Site do Webacula: <<http://webacula.sourceforge.net/>>.

Funcionalidades básicas:

- Restaurar um único job ou alguns arquivos de um job; arquivos a restaurar pesquisados com curingas e expressões regulares.
- Montar ou desmontar storages.
- Dashboard de jobs com erros.
- Status e informações dos volumes.
- Próximos jobs agendados.
- Jobs em execução.
- Últimos jobs terminados.
- Informação detalhada de volumes, pools, status e status dos clients.
- ACL gráfica.
- Diário de bordo para notas da equipe.
- Gráfico de Gantt dos jobs terminados.

- Repetir um job encerrado.

Requerimentos de instalação do Webacula

- Versão de Bacula compatível (por exemplo: 7.x).
- Zend Framework 1.10 ou posterior (eu realmente recomendo que você use a versão 1.11.5).
- PHP 5.2.4 ou superior, com a extensão PDO ativada.
- Apache com mod_rewrite.
- Pacote php-gd.
- Executar os scripts que criam as tabelas do Webacula e o usuário no banco de dados Bacula.

Resolvendo dependências e baixando o código do Webacula⁸¹

- Debian:

```
apt-get install apache2 php5 libapache2-mod-php5 php5-mysql php5-pgsql php5-gd
```
- CentOS:

```
yum install httpd php php-gd php-mysql php-pgsql
```

Vá para o diretório web do Apache. Por exemplo:

```
cd /var/www/
```

Faça o download do código webacula (tar.gz) e extraia para /var /www, assim como no comando a seguir, ou baixe outra versão através do site do projeto no Sourceforge através de seu navegador da internet (<<http://webacula.sourceforge.net/>>).

```
wget -T 3 -qO- \
http://downloads.sourceforge.net/project/webacula/webacula/7.0.0/
webacula-7.0.0.tar.gz \
| tar -xzf - -C /var/www
```

Em seguida, acesse o site do framework Zend (<<http://framework.zend.com/>>), baixe o pacote mínimo (Zend Framework 1.11.5 Minimal recomendado), extraia e copie recursivamente a **subpasta de nome library** para o diretório webacula criado (/var/www/webacula*/). Por exemplo:

```
wget -T 3 -qO- \
https://packages.zendframework.com/releases/ZendFramework-1.11.5/
ZendFramework-1.11.5-minimal.tar.gz \
| tar -xzf - --strip=1 */library/* \
-C /var/www/webacula-7*/
```

A árvore de diretórios deve ter a seguinte aparência: “/var/www/webacula.../library/ Zend...”, ou:

```
/var/www/webacula-7.0.0/
|- application
| |- controllers
| |- models
| `-- views
|- docs
|- install
|- html
|- languages
```

```
- library
.      |- Other
.      |- MyClass
.      |
.      ~- Zend (este vem do pacote mínimo do Zend)
.      |- Acl
.      |- Auth
.      |- Cache
.      |- Config
.      ...
```

Instalação do Webacula

Modifique a senha do usuário do administrador do banco de dados dentro do arquivo webaculainstall/db.conf:

```
cd /var/www/webacula-7*/install/
vi db.conf
# bacula settings
db_name="bacula"
# for Sqlite only
db_name_sqlite="/var/bacula/working/bacula.db" db_user="root"
# CHANGE_THIS
db_pwd="password" # <=(Altere para a senha de usuário do administrador do
banco de dados)

# Webacula web interface settings
...
#
# CHANGE_THIS
webacula_root_pwd="" <==(não altere isso agora).
```

Execute os seguintes scripts de acordo com o banco de dados correspondente. Se **MySQL**:

```
/var/www/webacula-7*/install/MySQL/10_make_tables.sh
/var/www/webacula-7*/install/MySQL/20_acl_make_tables.sh
```

Se **PostgreSQL** (você provavelmente precisará executar esses scripts com o usuário postgresql – por exemplo, **sudo -u postgres** antes do comando):

```
sudo -u postgres \
/var/www/webacula-7*/install/PostgreSQL/10_make_tables.sh
sudo -u postgres \
/var/www/webacula-7*/install/PostgreSQL/20_acl_make_tables.sh
```

Atualizar a senha do root Webacula

Vá para o seguinte diretório e crie um hash para uma nova senha de usuário administrador do Webacula (root):

```
cd /var/www/webacula-7*/install/
php ./password-to-hash.php nova_senha
```

Tome nota do hash MD5 gerado em um bloco de notas ou apenas copie. Exemplo de valor gerado: **\$P\$BeTUuIOyWSQI9C5ytE**.

•MySQL:

Acesse o console do MySQL:

```
mysql -u root -p
```

Use a base de dados do Bacula:

```
use bacula;
```

Agora, atualize o usuário root do Webacula para o criado:

```
update webacula_users set pwd='P$BeTUuIOyWSQl9C5ytE' where id=1000;
```

Ele deve retornar esta mensagem:

```
Query OK, 0 rows affected (0.00 sec)
```

```
Rows matched: 1 Changed: 0 Warnings: 0
```

Saia do MySQL (quit).

•PostgreSQL:

–su postgres.

–Acesse o console do PostgreSQL (por exemplo: psql).

–Conecte-se ao banco de dados Bacula. Ex.: “\connect bacula”.

–Atualize a senha do usuário root do Webacula:

```
update webacula_users SET pwd = 'P$BeTUuIOyWSQl9C5ytE' where id =  
'1000';
```

Configuração do Bacula para o Webacula

Alterar as opções do recurso Message padrão (standard) em **/etc/bacula/bacula-dir.conf**.

Deve ficar como a linha em negrito:

```
Messages {  
Name = Standard  
...  
catalog = all, !skipped, !saved  
...  
}
```

Reinicie o Bacula Director. Por exemplo:

```
/etc/init.d/bacula-director restart
```

Configuração do PHP

Altere os seguintes valores no arquivo **/etc/php5/apache2/php.ini**:

```
memory_limit = 32M  
max_execution_time = 3600
```

Configuração do Apache

Copie o arquivo de configuração modelo do Webacula para o diretório de configurações do Apache⁸²:

```
cp /var/www/webacula*/install/apache/webacula.conf /etc/apache2/sites-enabled/
```

Agora, vamos editar o arquivo **webacula.conf**:

```
vi /etc/apache2/sites-enabled/webacula.conf
```

Altere o alias e o caminho do site de acordo com suas necessidades. Por exemplo:

```
Alias /webacula /var/www/webacula-7.0.0/html  
<Directory /var/www/webacula-7.0.0/html>
```

Você também pode querer alterar as regras allow/deny etc. No final, ele deve ser terminar parecendo com isso:

```
Alias /webacula /var/www/webacula-7.0.0/html
```



```
<Directory /var/www/webacula-7.0.0/html>
RewriteEngine On
RewriteBase /webacula
RewriteCond %{REQUEST_FILENAME} -s [OR]
RewriteCond %{REQUEST_FILENAME} -l [OR]
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^.*$ - [NC,L]
RewriteRule ^.*$ index.php [NC,L]
php_flag magic_quotes_gpc off
php_flag register_globals off
Options Indexes FollowSymLinks
AllowOverride All
Order deny,allow
Allow from all
Allow from localhost
Allow from ::1
#
# change the settings below
#
# Allow from
</Directory>
```

Ative o módulo rewrite do Apache e reinicie o serviço. No Bash:

```
echo rewrite | a2enmod && service apache2 restart
```

Alterando permissões e configurações finais do Webacula

Forneça permissões para o usuário apache nas pastas do Webacula (substitua **www-data** por **apache** se estiver utilizando o CentOS):

```
chown -R www-data. /var/www/webacula-7*
```

Edite o arquivo de configuração do Webacula:

```
vi /var/www/webacula/application/config.ini
```

Altere o tipo de banco de dados (etc.) se necessário e a senha para acesso ao banco de dados do Bacula:

```
db.adapter = PDO_MYSQL # ou PDO_PGSQL
db.config.host = localhost
db.config.username = root
db.config.password = <password>
db.config.dbname = bacula
```

Para Debian 8, substitua também como segue:

```
bacula.sudo = ""
bacula.bconsole = "/usr/sbin/bconsole"
bacula.bconsolecmd = "-n -c /etc/bacula/bconsole.conf"
```

Salve e saia.

Vamos alterar algumas outras permissões para conceder acesso do Webacula ao bconsole:

```
chown www-data /usr/sbin/bconsole
chmod u=rwx,g=rx,o= /usr/sbin/bconsole
chown www-data /etc/bacula/bconsole.conf
chmod u=rw,g=r,o= /etc/bacula/bconsole.conf
chown www-data /etc/bacula
```

(Se estiver usando o CentOS, é necessário também desabilitar o **selinux** e fornecer permissões de **firewall** para o acesso externo do Webacula).

Feito! No seu navegador, digite o endereço da máquina Webacula mais o alias webacula para acessá-lo, como o exemplo.

http://endereço_ip/webacula

Observação: se você encontrar uma página em branco ou uma página de erro, leia também as seções a seguir.

Erro conhecido do Webacula: incompatibilidade com a versão do banco de dados do Bacula!

O Webacula, por padrão, procura uma versão específica do banco de dados do Bacula. Portanto, você pode encontrar esse erro no **log do Apache**:

```
[Sat Oct 27 16:54:41 2012] [error] [client 127.0.0.1] PHP Fatal error:
Uncaught exception 'Zend_Exception' with message 'Bacula version mismatch for
the Catalog database. Wanted 12, got 14. ' in /var/www/webacula-5.5.1/html/
index.php:186\nStack trace:\n#0 {main}\n thrown in /var/www/webacula-5.5.1/html/index.php on line 186
```

Para resolver isso você pode precisar alterar a versão esperada em **/var/www/webacula*/html/index.php**, linha 22. Este é o exemplo para a versão do banco de dados Bacula 15:

```
define('BACULA_VERSION', 15); // Bacula Catalog version
```

11.3 Webmin (permite alterar graficamente as configurações do Bacula)



Figura 11.4. Logotipo do Webmin.

Webmin é uma poderosa interface web desenvolvida em Perl para administração de servidores e possui suporte nativo para o Bacula.

Uma vantagem interessante é a capacidade de editar as configurações do Bacula graficamente. Foi provavelmente a primeira ferramenta de código aberto a trazer essa funcionalidade.

Instalação do Webmin

Para este manual usei sistemas operacionais Debian 8 e CentOS 7. Vamos resolver dependências⁸³:

•Debian:

```
apt-get install libnet-ssleay-perl openssl libauthen-pam-perl libio-pty-perl python apt-show-versions libdbd-mysql-perl libpg-perl libdbd-pg-perl
```

- CentOS:

```
yum -y install perl perl-Net-SSLeay openssl perl-IO-Tty perl-DBD-MySQL  
perl-DBD-Pg
```

Vá para o site webmin para buscar o pacote mais recente para sua distribuição (<<http://www.webmin.com/download.html>>) ou simplesmente faça o download dos pacotes com o **wget**, no terminal.

- Debian:

```
cd /tmp  
wget -T 3 \  
http://prdownloads.sourceforge.net/webadmin/webmin\_1.810\_all.deb  
  
dpkg -i webmin_1.810_all.deb
```

- CentOS:

```
cd /tmp  
wget -T 3 \  
http://prdownloads.sourceforge.net/webadmin/webmin-1.810-1.noarch.rpm  
  
rpm -Uivh webmin-1.810-1.noarch.rpm
```

- Para as duas distribuições: acesse o serviço Webmin⁸⁴ em https://endereço_ip:10000

Use o mesmo usuário do administrador do sistema operacional para autenticação – por exemplo, **root**.

Desça ao módulo Webmin Bacula usando o menu do lado direito: **System > Bacula Backup System**. Você deve configurar o acesso ao serviço de banco de dados do catálogo do Bacula:

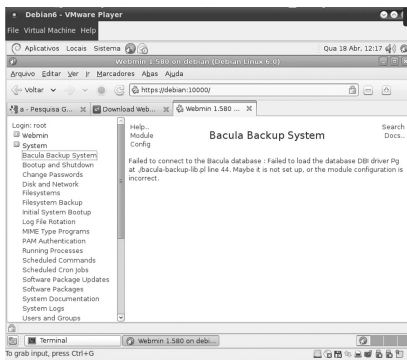


Figura 11.5. Configuração da primeira vez do módulo Webmin do Bacula.

Clique na configuração do módulo e altere suas credenciais de banco de dados corretamente:

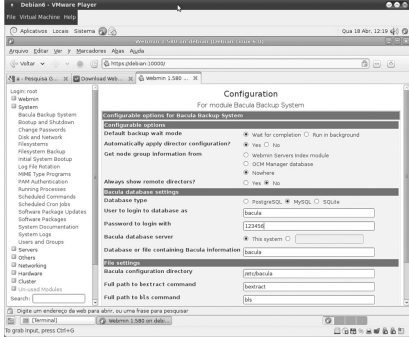


Figura 11.6. Formulário de configuração do módulo Webmin do Bacula.

Se tudo estiver certo, você poderá ver os ícones de configuração e outras operações na mesma página **System > Bacula Backup System**.

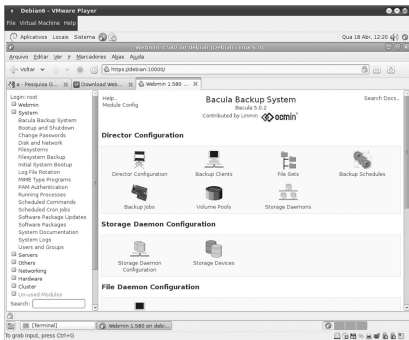


Figura 11.7. Painei do módulo Webmin do Bacula.

Apenas como exemplo, eu abri a opção de adicionar clientes na configuração do Director:

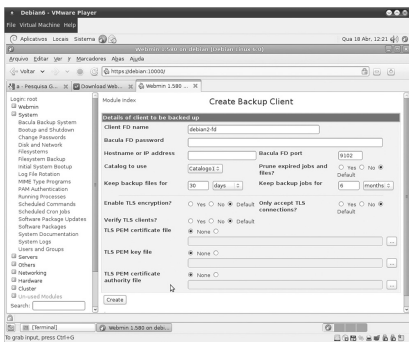


Figura 11.8. Adição de configuração do Webmin do cliente Bacula.

Também é possível enviar tarefas de backup e restauração, gerenciar volumes e verificar o status do Director na parte inferior da página do módulo principal:

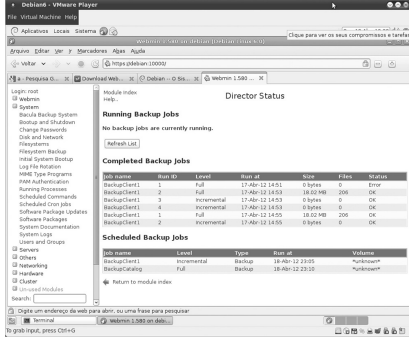


Figura 11.9. Módulo Webmin do Bacula, visualização Status do Director.

11.4 Baculum: mais uma interface de configuração e administração

Baculum é uma GUI que permite alterar as configurações do Bacula, gerenciar mídia, iniciar backup, restaurar etc.

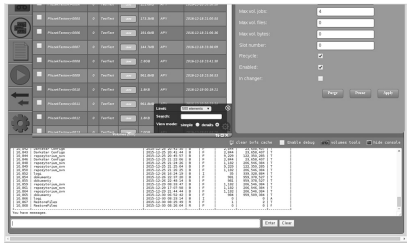


Figura 11.10. Guia de atributos de volume do Baculum.

•Certifique-se de que as dependências estão instaladas (Debian 8):

```
apt-get install apache2 libapache2-mod-php5 php5 php5-pgsql php5-mysql
php5-curl php5-json
```

•Pegue o pacote bacula-gui do código-fonte do Bacula ou baixe diretamente com o comando a seguir. Extraia a pasta **baculum** para **/var/www** ou o diretório da web de sua preferência.

```
wget -T 3 -qO- \
http://downloads.sourceforge.net/project/bacula/bacula/7.0.5/bacula-
gui-7.0.5.tar.gz \
| tar -xzf - --strip=1 */baculum/* -C /var/www/
```

•Edite e insira a seguinte linha em **/etc/sudoers** (usuário Apache⁸⁵):

```
vi /etc/sudoers
```

```
www-data ALL= NOPASSWD: /usr/sbin/bconsole
```

•Forneça permissões de gravação em pastas:

```
chown www-data /var/www/baculum/assets
chown www-data /var/www/baculum/protected/Data
chown www-data /var/www/baculum/protected/runtime
```

•Crie as credenciais para a autenticação do Apache. Exemplo com usuário admin⁸⁶:

```
htpasswd -c /etc/apache2/passwords admin
```

- Remova o arquivo padrão do Apache e crie um arquivo baculum.conf:

```
rm /etc/apache2/sites-enabled/000-default
vi /etc/apache2/sites-enabled/baculum.conf
```

Exemplo de configuração:

```
<VirtualHost *:80>
ServerName bacula
DocumentRoot /var/www/baculum

<Directory /var/www/baculum>
AllowOverride All
AuthType Basic
AuthName MyPrivateFile
AuthUserFile /etc/apache2/passwords
Require valid-user
</Directory>
</VirtualHost>
```

- Ative o Apache mod_rewrite:

```
echo rewrite | a2enmod
```

- Certifique-se de que o link existe⁸⁷:

```
ln -s /etc/apache2/mods-available/rewrite.load \
/etc/apache2/mods-enabled/rewrite.load
```

- Edite o php.ini:

```
vi /etc/php5/apache2/php.ini
```

- Altere o valor da diretiva error_reporting:

```
error_reporting = E_ALL; ~E_STRICT
```

- Reinicie o Apache:

```
/etc/init.d/apache2 restart
```

- Acesse o baculum: https://endereço_ip/baculum.

- Proceda com o assistente de instalação. A credencial de usuário a ser inserida deve corresponder à criada para o Apache anteriormente.

11.5 Bacula-web (interface de relatórios e estatísticas)

O Bacula-web é uma ferramenta de relatório e monitoração estática baseada na web (que não opera ou muda as configurações do Bacula) que fornece informações úteis sobre a infraestrutura de backup como jobs, pools, volumes, logs de tarefas, clientes e relatórios de jobs de backup. É lançado sob os termos da licença GPL v 2.0.

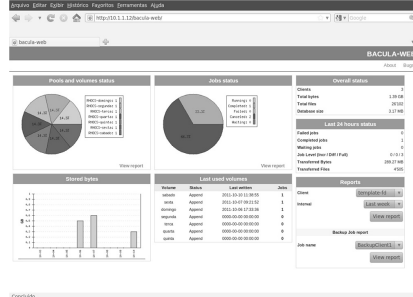


Figura 11.11. Um dos painéis do Bacula-web.

A GUI (interface gráfica de usuário) Bacula-web é principalmente escrita em PHP e é compatível com os sistemas operacionais preferidos, como RedHat, CentOS, Fedora, Ubuntu, Debian, Gentoo, FreeBSD etc.

Essas são algumas funcionalidades⁸⁸:

- **Página de relatório de jobs:** mostra os últimos 25 trabalhos do Bacula com várias opções de filtragem. Outra opção útil permite que você veja os logs de cada trabalho na página do relatório.
- **O relatório de backup do cliente** fornece, para cada cliente do Bacula, os detalhes a seguir:
 - SO, arquitetura e versão do cliente.
 - As últimas tarefas de backup conhecidas concluídas.
 - Gráfico dos últimos sete dias com tamanho e quantidade de arquivos armazenados.
- **Relatório de jobs de backup** exibe informações úteis sobre jobs do Bacula, como:
 - Últimos sete dias de bytes e arquivos armazenados por job.
 - Últimos jobs completos.
 - Gráficos dos últimos sete dias de bytes e arquivos armazenados por job.
- **Pools e volumes** fornecem uma lista de todas as pools do Bacula e volumes atribuídos com detalhes como nome do volume, bytes, tipo de mídia, data de vencimento, última data escrita, status.
- **A página de teste** fornece algumas informações úteis sobre a instalação e configuração do Bacula-web.

Instalação do Bacula-Web

Resolvendo dependências.

- Debian:

```
apt-get -y install apache2 libapache2-mod-php5 php5-mysql php5-gd php5-pgsql
```

- CentOS:

```
yum install httpd php php-gd php-gettext php-mysql php-pgsql php-pear-DB -y
```

Ainda para o **CentOS**, ative o início do serviço do Apache no momento da inicialização:

```
chkconfig httpd on
```

```
• Faça o download da última versão da Bacula-web:
mkdir /var/www/html/bacula-web
wget -T 3 -qO- \
http://www.bacula-web.org/download.html?file=files/bacula-web.org/
downloads/bacula-web-7.3.0.tgz \
| tar -xzf - -C /var/www/html/bacula-web
```

Edite o arquivo config.php do aplicativo:

```
cd /var/www/html/bacula-web/application/config
cp config.php.sample config.php
vi config.php
```

Altere de acordo com sua linguagem e configuração de conexão de banco de dados Bacula:

```
<?php$config['language'] = 'en_EN';

#MySQL bacula catalog
$config[0]['label'] = 'Backup Server';
$config[0]['host'] = 'IP';
$config[0]['login'] = 'db_user';
$config[0]['password'] = 'db_password';
$config[0]['db_name'] = 'bacula';
$config[0]['db_type'] = 'mysql';
$config[0]['db_port'] = '3306';?>
```

Salve e saia do arquivo.

Altere algumas permissões⁸⁹:

```
chown -Rv www-data:www-data /var/www/html/bacula-web
chmod -Rv u=rx,g=rx,o=rx /var/www/html/bacula-web
chmod -v ug+w \
/var/www/html/bacula-web/application/view/cache
```

Reinicie o Apache:

```
service httpd restart
```

Execute o seu primeiro acesso: http://endereço_ip/bacula-web

11.6 Outros projetos

- Bweb – <http://www.baculasystems.com/>. A melhor e mais completa interface de gestão, operação e configuração do Bacula. Ele reúne a maioria dos recursos de outras GUIs apresentadas e mais (por exemplo: autenticação de diretório de serviço, agrupamento de clientes etc.). **No entanto, ele só está disponível para a versão Enterprise do Bacula.**
- Jbacula – <https://sourceforge.net/projects/jbacula/>. Permite alterar arquivos de configuração do Bacula. Provavelmente descontinuado.
- Ddrescue – <http://www.gnu.org/software/ddrescue/ddrescue.html>. Tenta recuperar dados de mídias com defeito.
- Bacuview. Um monitor Ruby para o Bacula.

⁸⁹ Quanto ao CentOS/RedHat, você provavelmente encontrará o código necessário no pacote qwt-devel e os outros equivalentes, disponíveis em alguns repositórios não padrão (por exemplo, EPEL): http://www.rackspace.com/knowledge_center/article/install-epel-and-additional-repositories-on-centos-and-red-hat.

- 81 Isso abrange a implantação para MySQL e PostgreSQL do catálogo do Bacula.
- 82 No CentOS essa pasta deve estar em: **/etc/httpd/**.
- 83 Eu tentei cobrir as dependências do catálogo do Bacula para MySQL e PostgreSQL.
- 84 Observação: para recursos de restauração, o módulo webmin do Bacula precisa do plug-in do navegador Java. Se você usa Debian Linux, pode usar a versão aberta – por exemplo, apt-get install icedtea-7-plugin.
- 85 Tipicamente, **apache** na família CentOS.
- 86 Nota: O “c” só deve ser usado para criar novos usuários.
- 87 Lembre-se de que é uma única linha (a barra invertida apenas a quebra).
- 88 Do site do desenvolvedor: <http://docs.bacula-web.org/en/master/01_about/features.html>.
- 89 O usuário do Apache em alguns CentOS pode ser www-data.

12

Comandos do Bacula

12.1 Comandos mais importantes do Bacula

Cancel

```
cancel [jobid=<number> job=<job-name> ujobid=<unique-jobid> all]
```

Cancela um job de maneira segura. Você pode especificar um JobID, todos os jobs com a palavra-chave **all** ou simplesmente digitar **cancel** (sem argumentos) e selecionar um trabalho para cancelar a partir de um prompt. Quando um trabalho é cancelado, pode levar algum tempo para o Bacula terminar suas operações internas e efetivamente cancelá-lo (você pode verificar isso com o comando **status director**, na sessão **Running Jobs**).

Este comando teve sua **importância diminuída** desde o advento do Bacula 7.4, que trouxe o comando **stop**. É possível **resumir** um job: reiniciar a partir do ponto onde foi parado com o **stop**. No entanto, **não é possível retomar uma tarefa cancelada**.

Delete

```
delete [volume=<vol-name> pool=<pool-name> job jobid=<id>]
```

O comando delete pode apagar registros do catálogo do Bacula de um volume, uma pool ou um job do catálogo e informações associadas. Este comando opera somente no catálogo e não afeta os volumes escritos no storage. Não respeita os tempos de retenção e pode ser perigoso. Normalmente é feito quando os volumes físicos são perdidos (por exemplo: uma fita com defeito) ou quando você executa vários jobs de teste e quer limpá-los da lista de jobs. Você deve usá-lo também se quiser apagar uma pool não mais utilizada que você já tenha apagado do arquivo de configuração bacula-dir.conf.

Exemplos:

```
delete pool=<pool-name>
delete volume=<volume-name> pool=<pool-name>
delete JobId=<job-id> JobId=<job-id2>
delete Job JobId=n,m,o-r,t ...
```

Disable

```
disable job<job-name>
```

Este comando permite desativar um job da Agenda (Schedule), enquanto o Bacula Director não é recarregado com o comando **reload** ou tem seu daemon reiniciado. Uma maneira mais permanente de desativar um job da programação é usar a diretiva **Enabled = no** no recurso

Job. Os trabalhos desativados ainda podem ser executados manualmente com o comando **run**.

Enable

```
enable job<job-name>
```

É o oposto do **disable**. Reativa um job desabilitado.

Estimate

```
estimate job=<job-name> listing client=<client-name> fileset=<fileset-name>  
level=<level-name>
```

O “**estimate listing**” é um dos comandos mais importantes do Bacula. Ele pode ser usado para **simular a execução de um job de backup**, exibindo os arquivos que provavelmente seriam copiados, a quantidade e seu tamanho total. É útil para testar FileSets e configurações de jobs antes de enviar um backup real. Uma estimativa bem-sucedida é uma boa evidência de que esta tarefa de backup terminará em OK. Você também pode estimar uma tarefa de backup para um determinado nível: “level = Incremental”, “level = Differential” etc.

O tamanho total de backup é uma estimativa e pode mudar se a compactação for ativada (por exemplo) e por outros fatores. Exemplo:

```
estimate job=NomeJobBackup listing
```

```
...  
drwxr-xr-x 2 root root 4096 2016-12-26 10:43:20 /etc/rc0.d  
-rw-r--r-- 1 root root 653 2016-03-04 21:43:45 /etc/chatscripts/pap  
-rw-r----- 1 root dip 656 2016-03-04 21:43:45 /etc/chatscripts/provider  
-rw-r--r-- 1 root root 950 2016-03-04 21:43:45 /etc/chatscripts/gprs  
drwxr-s--- 2 root dip 4096 2016-04-26 11:30:51 /etc/chatscripts  
-rw----- 1 root root 0 2016-03-04 21:46:18 /etc/subuid-  
drwxr-xr-x 169 root root 12288 2017-01-03 09:54:02 /etc  
2000 OK estimate files=4,345 bytes=9,124,748
```

*

Se você não usar a palavra-chave de **listing**, a estimativa apenas exibirá o total de arquivos e o tamanho dos backups estimados.

Observação: se a saída de um **estimate listing** for muito grande para o seu fluxo de terminal, você pode salvar a saída em um arquivo com um redirecionamento do Bacula.

Por exemplo:

```
@output /tmp/listing  
estimate job=AnyBackupJob listing level=Full  
@output
```

Help

```
help | help command
```

Mostra todos os comandos disponíveis no bconsole. Você também pode usar **help nome_do_comando** para buscar todas as opções do comando possíveis, muito útil para criar comandos silenciosos para o Bacula.

Label

Este comando cria manualmente um novo volume do Bacula, na mídia física e no catálogo do Bacula. Esta é a forma completa:

```
label storage=<storage-name> volume=<volume-name> slot=<slot>
```

Normalmente, o comando **label** sozinho não é muito usado. Para discos e unidades de fita manuais, o **label automático** é preferível⁹⁰. Para bibliotecas de fita que possuem **leitores de código de barras** e **etiquetas físicas** devidamente fixadas, o comando para criar volumes para todas as fitas novas (virgens) é o **label barcodes** (bconsole). O Bacula irá inseri-las uma por uma e rotulá-las logicamente.

Vários **caracteres especiais** são proibidos no nome do volume. Além disso, você pode receber alguns alertas de IO ao rotular novas fitas, mas isso é aceitável. É apenas uma maneira de dizer que não há dados em fita.

O comando label pode falhar por alguns motivos:

- a)O novo volume é semelhante ao já existente e deve ser único.
- b)Já existe uma fita montada rotulada no dispositivo na unidade manual.
- c)Não há fita na unidade manual.
- d)O Bacula não pode acessar o caminho para o Archive Device no caso de armazenamento em disco (normalmente a permissão de escrita no diretório não foi concedida).

Ao rotular volumes para os jobs de backup aguardando intervenção, verifique se eles são criados na pool definida. **Um erro comum é esperar que o Bacula escreva um job de backup em um volume que pertence a uma pool diferente daquela a que o job foi submetido. Isso nunca acontecerá.**

Se a fita já estiver sendo usada por outro software ou você simplesmente quiser descartar todos os conteúdos e o nome anterior do volume, você pode apagá-los rapidamente com o seguinte comando (a fita já deve estar inserida na unidade):

```
mt -f /dev/st0 rewind mt -f /dev/st0 weof
```

Em seguida, você pode fazer a operação de **label**.

List

O comando **list** obtém muitas informações do catálogo Bacula. Existem vários usos úteis:

```
list jobs
```

Lista o estado atual de todos os jobs em execução e terminados para esse ambiente Bacula. Você pode usar isso para exibir manualmente e selecionar o JobId para uma restauração, ou apenas verificar os trabalhos concluídos. Existem códigos para o tipo de job, nível e status, que são detalhados no **Anexo I** deste livro.

Dica: o Bacula 7.4 traz novas palavras-chave de listagem que são ótimas para uso com o comando **list jobs**:

•Opções de ordenação:

```
ordem = asc | ascendente, ordem = desc | descendente
```

- Limitando o tamanho da lista:

```
limit=nn
```

- Filtrar um determinado status de término do trabalho (por exemplo, cancelado):

```
list jobs jobstatus=A
```

- Filtrar apenas para um determinado cliente:

```
list jobs client=inesbr-fd
```

- Exibir apenas trabalhos com erros:

```
list jobs joberrors
```

Outro uso do **list** comum é recuperar rapidamente o log de um trabalho específico terminado:

```
list joblog jobid=<id>
```

Também é possível listar todos os arquivos copiados de um determinado job de backup. **Não seja tímido – experimente!**

```
list files jobid=<id>
```

Existem ainda os comandos **list media** (ou **list volumes**), que são muito necessários. Eles imprimem uma tabela com as informações essenciais dos volumes de backup. Por exemplo:

```
*list media pool=File
```

MediaId	VolumeName	VolStatus	Enabled	VolBytes	VolPfiles	VolRetention
1	Vol-0001	Append	1	17,749,249,834	4	31,536,000
2	Vol-0002	Append	1	210	0	31,536,000
3	Vol-0003	Append	1	210	0	31,536,000
4	Vol-0004	Append	1	210	0	31,536,000

Figura 12.1. List media de uma pool (primeira metade da tela)⁹¹.

Recycle	Slot	InChanger	MediaType	LastWritten	ExpiresIn
1	0	0	File1	2016-08-04 09:41:37	30,326,721
1	0	0	File1	NULL	NULL
1	0	0	File1	NULL	NULL
1	0	0	File1	NULL	NULL

Figura 12.2. Lista media de uma pool (segunda metade da tela).

• **MediaId:** identificador único do volume no Bacula. Ele pode ser usado para selecionar volumes rapidamente para outros comandos (purge, delete etc.).

• **VolumeName:** também é um identificador de volume exclusivo, mas definido pelo usuário. Existem alguns caracteres especiais que são proibidos para nomes de volume. Uma vez que eles são criados automaticamente para discos, é possível definir nomes com variáveis (por exemplo: nome da pool, data etc.). No caso de robôs de fitas, o VolumeName deve ser o mesmo fornecido pelo código de barras das fitas.

• **VolumeStatus:** indica o estágio do ciclo de vida em que o volume do Bacula está. Por exemplo: se o Bacula pode selecioná-lo para escrever novos dados, se ele pode ser substituído (reciclado) etc. – tudo de acordo com o seu algoritmo de seleção automática de volume⁹² e demanda de armazenamento de backup. Esses são os possíveis estados:

– **Append:** é o primeiro status de volume quando criado ou reciclado. Isso significa que você pode continuar anexando dados a ele, uma vez que é uma sequência de bytes. Não é possível apagar ou substituir os dados escritos, a menos que você faça um purge/recycle de todo o volume. Enquanto o volume estiver com esse status, cada vez que receber novos jobs de backup ele terá a contagem de sua retenção de volume resetada, uma vez que deve garantir que os novos jobs serão mantidos para o período

de tempo determinado na pool.

–**Full**: significa que a mídia física ficou cheia (no caso a fita ou partição de disco) ou atingiu seu tamanho máximo (`MaximumVolumeBytes`, definido no caso apenas para volume em disco). Dessa forma, você provavelmente não pode adicionar mais dados nela. Para gravá-la novamente você deve reciclá-la ou purgá-la, o que significa que você deve descartar todos os dados já gravados sobre ele.

–**Used**: neste status você provavelmente teria espaço para escrever mais jobs no volume, mas você quer ter certeza de que o cronômetro de retenção de volume rode sem interrupções e o volume seja reciclado em um tempo previsível. Você pode definir limitadores de uso no recurso Pool para que o Bacula transfira automaticamente o status do volume para o usado (por exemplo: `VolumeUseDuration`, `MaximumVolumeJobs`).

–**Error**: por algum motivo, o Bacula tentou acessar um volume de arquivo de fita ou disco e não conseguiu encontrá-lo; ou um erro de escrita física impediu o Bacula de escrever no volume. Você deve verificar os logs e as configurações para saber o que aconteceu, mas é geralmente seguro corrigir ou alterar um volume de erro para o status usado se ele tem dados e você deseja que passe pelo tempo de retenção; ou apenas purgá-lo para que o Bacula possa sobrescrevê-lo.

–**Recycled**: permite que os volumes de retenção expirados sejam reciclados e usados novamente por um próximo job de backup.

–**Archive**: este é um estado de conservação. A retenção de volume é ignorada e o volume nunca é reciclado. Você pode definir manualmente o volume com o status **archive** através do comando **update > volume parameters**, do bconsole. Mas você também pode encontrar esse status quando você usa bscan para recuperar uma informação de volume para o catálogo do Bacula, porque ele pode recuperar o status de volume real antigo.

–**Disabled**: o volume ficará indisponível para as rotinas do Bacula.

–**Cleaning**: as fitas de limpeza devem ter esse status. Normalmente é automaticamente dado pela diretiva **CleaningPrefix**, do recurso Pool.

–**Read-only**: o volume só pode ser lido, não escrito.

•**Enabled**: pode ter um dos seguintes valores: não – falso, sim – verdadeiro, arquivado (0, 1, 2, respectivamente). Os volumes arquivados não serão usados, nem o registro da mídia no catálogo será “prunado” (podado). Volumes desabilitados não serão usados para backup ou restauração.⁹³

•**VolBytes**: tamanho atual do volume de backup, em bytes.

•**VolFiles**: o Bacula escreve marcadores EOF (final de arquivo) em fitas para tornar a restauração mais rápida (ele usa esses marcadores para pular/avançar rapidamente sobre os arquivos que não precisa ao restaurar).⁹⁴ Esses são mais importantes para as unidades de fita, que são sequenciais. E o tamanho dos marcadores pode ser ajustado no arquivo de configuração do Bacula do Storage Daemon, diretiva `MaximumFileSize` (recurso Device do bacula-sd.conf).

•**VolRetention**: tempo de retenção de volume configurado, em segundos.

•**Recycle**: indica que o volume está configurado para ser reciclado automaticamente pelo Bacula se for possível: o tempo de retenção é excedido e não há outro volume melhor na mesma pool para escrever, de acordo com o algoritmo de seleção. Esse valor

é retirado da diretiva Recycle, recurso Pool, em bacula-dir.conf.

- Slot:** endereço da mídia da biblioteca de fitas (gaveta). Permite ao Bacula fazer toda a manipulação de fitas automaticamente. Para volumes de disco não é utilizado (sempre 0).

- InChanger:** é um simples campo verdadeiro ou falso (1 ou 0) e também é usado apenas para bibliotecas de fitas. Informa o Bacula (e o administrador do backup) se uma determinada fita estiver dentro ou fora da biblioteca. Para discos é inutilizado (sempre 0).

- MediaType:** cria uma matriz de compatibilidade entre mídias e dispositivos de backup (um disco ou uma unidade de fita). O atributo **MediaType** do dispositivo deve corresponder (exatamente) ao **MediaType** do volume; caso contrário, o Bacula achará que eles não são compatíveis e não poderão escrever ou ler. Isso pode ser útil para lidar com bibliotecas de fitas com **diferentes tipos de unidades (por exemplo: LTO4 e LTO5)** ou para tornar o Bacula ciente de que os volumes são armazenados **em diferentes discos ou pontos de montagem**.

- LastWritten:** última vez que um job de backup foi gravado nesse volume. Por falar nisso: o Bacula prefere continuar escrevendo no último volume que estava usando, até que ele esteja **full** ou qualquer outra condição configurada (por exemplo: **VolumeUseDuration**) para que seu status seja alterado para **used**. Em segundos.

- Expires in:** tempo de retenção remanescente, em segundos. Essa contagem é apenas honesta se o volume não pode ser mais escrito (por exemplo: estado **full** ou **used**); uma vez que o volume é escrito, este temporizador é reiniciado.

Felizmente, este é o final da explicação de saída de comando de **list media**. Mas ainda falando sobre o comando **list**, existem alguns outros usos que podem ser úteis:

```
list jobid=<id>
list ujobid<unique job name>
list job=<job-name>
list jobname=<job-name>
list jobmedia
list jobmedia jobid=<id>
list jobmedia job=<job-name>
list files jobid=<id>
list files job=<job-name>
list pools
list clients list jobtotals
list volumes
list volumes jobid=<id>
list volumes pool=<pool-name>
list volumes job=<job-name>
list volume=<volume-name>
list nextvolume job=<job-name>
list nextvol job=<job-name>
```

A opção “nextvolume” é usada para adivinhar qual será o volume que o Bacula irá pedir para um backup em um determinado dia. É apenas uma previsão, uma vez que o Bacula tem vários critérios para definir o volume que vai usar em um determinado momento, e as coisas podem mudar. Este é um exemplo de uma estimativa três dias à frente:

```
list nextvol job=MyJob days=3
```

Messages

messages

Cada mensagem nunca vista dos backups (as mesmas que vão para o log) é impressa na tela. Como a maioria dos comandos do Bacula, você pode apenas usar a abreviação “**m**” para produzir o mesmo resultado. As novas mensagens só podem ser vistas uma vez através do comando **messages**.

Mount

```
mount storage=<storage-name>
mount [ jobid=<id> | job=<job-name> ]
```

É um comando muito importante que tem finalidades diferentes de acordo com o dispositivo de armazenamento. Garante ao Bacula acesso a uma mídia removível (por exemplo: **unidade de fita manual**); força o Bacula a carregar uma fita específica em uma unidade de **biblioteca de fitas**; força o Bacula a procurar novamente os volumes disponíveis para backup quando o trabalho estiver aguardando intervenção (**backups de disco**).

Prune

```
prune files|jobs|volume client=<client-name> volume=<volume-name>
```

O comando **prune** afeta apenas as informações do catálogo do Bacula sobre jobs de backup, mais precisamente as informações das tabelas **Job** e **File** se estiverem fora de suas respectivas retenções (diretrizes **FileRetention** e **JobRetention** do **bacula-dir.conf**, como citado anteriormente).

Normalmente, o Bacula limpa essas informações automaticamente (**diretiva AutoPrune=yes no bacula-dir.conf**) ao final de cada trabalho de backup. No entanto, o administrador pode definir isso como não (no) e, ao invés, utilizar o comando **prune** manualmente para reduzir o tamanho do banco de dados do catálogo quando julgar necessário. Outro método usual é executar periodicamente o comando **prune** através de um “**Admin Job**” especial, agendado para uma hora diferente da execução do backup, para não usar recursos importantes da máquina ao mesmo tempo.

Purge

```
purge files jobs volume=<vol> [mediatype=<type> pool=<pool> allpools
storage=<st> drive=<num>]
```

O comando **purge** exclui todas as informações de um job, cliente ou sobre um volume, sem respeitar qualquer tempo de retenção. Geralmente é usado para reciclar um volume quando você precisa de espaço para armazenar novos backups.

```
purge volume|volume=<vol-name>
```

Purge é o comando usado se o administrador precisar forçar a reciclagem de um **volume**, mesmo que ainda esteja dentro do seu tempo de retenção. Por padrão, ele não vai acessar dados sobre o volume para excluir ou fazer qualquer modificação imediatamente, mas o Bacula irá sobrescrevê-lo se quando houver uma demanda de escrita por novos jobs de backups.

Outros usos:


```
purge files jobid=<jobid> (or) job=<job-name>  
purge jobs client=<client-name>
```

Release

```
release storage=<storage-name>
```

Este comando é usado para fazer com que o daemon de armazenamento rebobine e ejete a fita atual na unidade (para que seja retirada do robô, por exemplo) e para reler a etiqueta de volume na próxima vez que o drive for solicitado. Normalmente, é mais usado do que o comando **umount** se o desejo é retirar fitas do robô, por exemplo, pois o comando **release** não bloqueia o acesso futuro do Bacula ao drive no caso de nova demanda de gravação.

*Após um comando de liberação, o dispositivo ainda é mantido aberto pelo Bacula (a menos que “Always Open” esteja definido como “No” na configuração do daemon de armazenamento) para que ele não possa ser usado por outro programa. No entanto, com algumas unidades de fita, o operador pode remover a fita atual e inserir um diferente, e quando o próximo trabalho for iniciado o Bacula saberá reler o rótulo da fita para descobrir qual fita é montada. Se você quiser ser capaz de usar a unidade com outro programa (por exemplo, mt), você deve usar o comando **umount** para fazer com que o Bacula libere completamente o dispositivo.*⁹⁵

Reload

```
reload
```

O comando **reload** aplica mudanças de configurações que foram feitas para o **bacula-dir.conf** imediatamente, sem precisar reiniciar o **daemon do Director** que abortaria os jobs que estão sendo executados.

Se houver qualquer erro de sintaxe para o seu **bacula-dir.conf**, você deve verificar com o comando **message** do bconsole ou log Bacula. No entanto, o Director permanecerá em execução com sua configuração antiga.

Restore

O comando **restore** sempre levará a encontrar o trabalho ou trabalhos que serão necessários para uma restauração de arquivo. O usuário pode fazer isso de várias maneiras: é possível usar o **list jobs** no bconsole para selecionar manualmente os jobs (por exemplo: um completo e um diferencial) e inserir o jobid entre as vírgulas na opção de restauração manual (**menu de restauração # 3**) ou especificar um dado tempo no passado para o Bacula, e ele irá automaticamente selecionar os últimos jobs antes disso. Detalhes do uso **interativo da restauração** (sem especificar qualquer opção) serão discutidos no próximo capítulo deste livro.

No entanto, aprender a sintaxe de restauração completa permitirá ao administrador do Bacula **criar rapidamente** trabalhos de restauração com um comando de linha única, e talvez até mesmo automatizar o processo de restauração (por exemplo: para fins de teste). Um exemplo simples de restauração silenciosa de todos os arquivos seria:

```
restore jobid=9 all done yes
```

Mas existem várias outras opções que podem ser usadas:

```
restore where=</path> client=<client> storage=<storage> bootstrap=<file>  
restorejob=<job> comment=<text> jobid=<jobid> file=<file_to_restore> current  
client=<bkp-client> restoreclient=<restore-client> before <time-
```

```
specification> done select all
```

Aqui vamos nós⁹⁶:

- **all** – Marca todos os arquivos [dos jobs de backup selecionados] a serem restaurados.
- **select** – Usa o método interativo de seleção de árvore.
- **done** – Não solicita o usuário no modo árvore.
- **current** – Seleciona automaticamente o conjunto mais atual de backups para o cliente especificado (último full mais diferencial ou incrementais).
- **client=xxxx** – Inicialmente especifica o cliente do qual o backup foi feito (...).
- **restoreclient=xxxx** – Se a palavra-chave for especificada, a restauração será gravada nesse cliente [é claro que o cliente de restauração pode ser diferente do cliente de backup original].
- **jobid=nnn** – Especifica um JobId ou lista separada por vírgulas dos JobIds a serem restaurados.
- **before="YYYY-MM-DD HH:MM:SS"** – Especifica uma data e hora para a qual o sistema deve ser restaurado. Somente trabalhos iniciados antes da data/hora especificada serão selecionados: (...) o Bacula irá automaticamente encontrar o Full e todos os incrementais e diferenciais executados antes da data que você especificar (...).
- **file=filename** – Especifica um nome de arquivo a ser restaurado. Você deve especificar o caminho completo e o nome do arquivo. Prefixar a entrada com um sinal menor que (<) fará com que o Bacula presuma que o nome do arquivo está em seu sistema e contém uma lista de arquivos a serem restaurados. O Bacula, assim, lê a lista desse arquivo. Várias especificações de **file=xxx** podem ser utilizadas em uma mesma linha de comando [ex.: **file=</home/hfaria/restore_list**, e este arquivo contém uma lista com arquivos do backup que devem ser selecionados para essa restauração. É possível recuperar a lista completa de arquivos para um determinado job com o comando **list files jobid=xx**].
- **jobid=nnn** – Especifica um JobId a ser restaurado.
- **pool=pool-name** – Especifica um nome de pool a ser usado para seleção de volumes ao especificar as opções 5 e 6 (restaurar o sistema atual e restaurar o sistema atual antes da data especificada). Isso permite que você tenha várias pools, possivelmente um off-site, e selecionar uma para a restauração.
- **where=[path]** – Os arquivos são restaurados no diretório especificado pelo where [lembre-se: este caminho é criado no cliente de restauração. Por padrão, é o mesmo cliente do qual o backup foi feito].
- **yes** – Executa automaticamente a restauração sem solicitar modificações (mais útil em scripts de lote) [permite que a restauração silenciosa seja executada].
- **strip_prefix=/prod** – Remove uma parte do nome do arquivo ao restaurar.
- **add_prefix=/test** – Adiciona um prefixo a todos os arquivos ao restaurar (similar ao where=) (mas não pode ser usado com where=).
- **add_suffix=.old** – Adiciona um sufixo a todos os seus arquivos.
- **regexwhere=!a.pdf!a.bkp.pdf!** – Faz a manipulação de nome de arquivo complexa como com comando sed unix. Substituirá a manipulação de outro nome de arquivo.

Resume

```
resume
```

```
incomplete job=<job-name> client=<client-name>
fileset=<FileSet-name> level=<level-keyword>
storage=<storage-name>when=<time-specification>
comment=<text> spooldata=<bool> jobid=<jobid>
```

Este comando foi introduzido no Bacula 7.4 e permite ao usuário reiniciar jobs interrompidos a partir do ponto em que foram descontinuados. Os jobs que **falham** devido a alguns erros específicos ou deliberados interrompidos pelo utilizador (com o comando **stop**) são elegíveis para ser retomados.

Exemplo de uso:

```
*resume
```

```
Automatically selected Catalog: MyCatalog
```

```
Using Catalog "MyCatalog"
```

```
You have the following choices:
```

```
1: Incomplete
```

```
2: Canceled
```

```
3: Failed
```

```
4: All
```

```
Select termination code: (1-4): 1
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+
| JobId | Name | StartTime | Type | Level | JobFiles | JobBytes | JobStatus |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+
| 20 | BkpArquivosBacula | 2016-02-26 18:04:37 | B | F | 0 | 0 | I |
| 22 | BkpArquivosBacula | 2016-02-26 18:10:20 | B | F | 2,501 | 32,722,228 | I |
| 25 | BkpArquivosBacula | 2016-02-26 18:12:44 | B | F | 628 | 376,861 | I |
| 27 | BkpArquivosBacula | 2016-02-26 18:13:25 | B | F | 2,685 | 106,706,580 | I |
| 28 | BkpArquivosBacula | 2016-02-26 18:16:11 | B | F | 4,043 | 138,253,969 | I |
| 29 | BkpArquivosBacula | 2016-02-26 18:17:34 | B | F | 2,504 | 35,515,855 | I |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+
Enter the JobId list to select: 29
```

```
Job queued. JobId=29
```

Exemplo em uma linha:

```
resume incomplete jobid=29
```

Run

Este comando permite iniciar tarefas de backup ad hoc, a qualquer momento:

```
run job=<job-name> client=<client-name> fileset=<FileSet-name> level=<level-
keyword> storage=<storage-name> where=<directory-prefix> when=<universal-
time-specification> yes
```

Exemplo de execução de uma tarefa de backup em uma linha:

```
run job=BackupBaculaConfs level=full yes
```

Usando o comando **run** sozinho, será aberto um prompt para selecionar o trabalho configurado para executar e de onde é possível alterar qualquer um dos atributos, também

interativamente. Por exemplo:

A job name must be specified. The defined Job resources are:

- 1: BackupBaculaConfs
- 2: BackupDNS
- 3: BackupFileServer
- 4: BackupWindowsAD
- 5: BackupBaculaCatalog

Select Job resource (1-5):

Se 5 for selecionado:

Run "backup" job

JobName: BackupBaculaCatalog

FileSet: Catalog

Level: Incremental

Client: bacula-fd

Storage: DLTDrive

Pool: Daily

When: 2014-04-23 17:08:18

OK to run? (yes/mod/no):

Aqui você pode usar a palavra **mod** para modificar qualquer um dos padrões de jobs que são dados pelos recursos **Job + JobDefs** no **bacula-dir.conf**:

Parameters to modify:

- 1: Level
- 2: Storage
- 3: Job
- 4: FileSet
- 5: Client
- 6: When
- 7: Pool

Select parameter to modify (1-7):

Depois de fazer as alterações necessárias, basta digitar **yes** para iniciar o job de backup. É aconselhável acompanhar as primeiras tarefas com o comando **messages** e **status director**, a fim de garantir que o job esteja realmente funcionando e não esperando por restrição indesejada (por exemplo: não há volumes graváveis ou recicláveis na pool etc.).

Show

```
show job=<xxx> | pool=<yyy> | fileset=<aaa> | schedule=<sss> | client=<zzz> |  
storage=<sss> | disabled | all
```

O comando **show** mostrará como seu bacula-dir.conf está configurado para cada recurso selecionado. Por exemplo:

*show filesets

```
FileSet: name=libreoffice  
    O M  
    N  
    I /home/hfaria/Documents  
    N  
FileSet: name=Catalog  
    O M
```

```
N
I /var/lib/bacula/bacula.sql
N
```

Este comando é muito útil para verificar como o seu Bacula Director está configurado, especialmente para solucionar problemas e pedir suporte.

Status

```
status [all | dir=<dir-name> | director | client=<client-name> |
storage=<storage-name> | days=nnn]
```

O comando de status é muito importante não apenas para buscar informações de job de backup em execução a partir da perspectiva de cada daemon do Bacula, mas também para **testar** a conexão entre o Director e os clients e storages recentemente criados. Um status bem-sucedido do cliente e do armazenamento é uma boa indicação de que a tarefa de backup será capaz de ser executada sem problemas.

Se você optar por executar um “**status director**”, você poderá ver os jobs agendados para as próximas **24 horas** e os trabalhos em execução com seu respectivo status (em execução, aguardando intervenção etc.).

Sobre os jobs agendados, ele também exibe uma estimativa de qual volume o Bacula provavelmente irá escolher para começar a escrever para esses jobs, de acordo com seu algoritmo. Se você tiver apenas volumes na Pool Scratch ou tiver configurado o label automático de volumes, o Bacula pode não ser capaz de prever o volume e apenas imprimir a palavra **unknown** (desconhecido) nesse campo. Não será necessariamente a razão dos backups programados com falha.

Ainda na seção **Running Jobs** (trabalhos em execução) da saída do **status director**, você pode ter um desses status:

```
16 Full CatalogBackup.2004-03-13_01.10.00 is waiting for higher priority jobs
to finish
13 Differential Minou.2004-03-13_01.05.09 is waiting on max Storage jobs
12 Full Rufus.2004-03-13_01.05.04 is running
```

O **Job** de **Id 12** está em execução. Você pode ver qual arquivo, o total de bytes e arquivos que estão sendo copiados com um novo comando **status client**, para o respectivo cliente.

JobId 13 está aguardando o **JobId 12** terminar, uma vez que provavelmente o **MaximumConcurrentJobs** é igual a 1 ou não definido no recurso Storage (bacula-dir.conf).

O **JobId 16 (CatalogBackup)** deve ter, e neste caso também tem, menor prioridade do que outros trabalhos, e também está esperando que terminem para iniciar.

Se o seu job parece estar parado, a coisa mais sensata a fazer é pedir um status do storage e olhar para os volumes que o Bacula pode estar tentando acessar:

```
status storage=File
...
Device status:
Autochanger "DDS-4-changer" with devices: "DDS-4" (/dev/nst0)
Device "DDS-4" (/dev/nst0) is not open.
Device is BLOCKED. User unmounted. Drive 0 is not loaded.
```

```
Device "Dummy" is not open or does not exist. No DEVICE structure.  
Device "DVD-Writer" (/dev/hdc) is not open. Device "File" (/tmp) is not open.  
Device is BLOCKED waiting for media.  
====  
...
```

A mensagem **"Device is BLOCKED waiting for media"** (dispositivo está **BLOQUEADO esperando por mídia**) pode referir-se a coisas diferentes: todos os seus volumes de fita estão sob o período de tempo de retenção e você não tem volumes novos ou com status **append** para escrever; em último caso, o Bacula só poderá escrever em uma fita reciclada que está fora de sua biblioteca de fitas, e neste caso ele está pedindo para que essa fita seja inserida; ou por algum motivo o Bacula não consegue montar o volume que deseja automaticamente no drive do robô de fitas.

Se você criar novos volumes ou purgar os antigos e o Bacula Storage ainda estiver bloqueado, você poderá utilizar um comando **mount**. Ele também força o Bacula a ler novamente as listas de volumes e tentar encontrar outro adequado para prosseguir com o backup.

Unmount

Para **drive de fitas manual** é um comando de uso obrigatório para liberar a fita após o uso do Bacula e posterior ejeção pelo sistema operacional. Essa atividade pode ser automatizada se você configurar sua fita com um dispositivo trocador virtual (ver capítulo sobre fitas magnéticas); para **bibliotecas de fitas** ele pode ser usado para descarregar fitas das unidades, especificando o drive onde a fita está e a gaveta para onde deve ir (slot); para **armazenamento em disco** desconheço uso para este comando.

```
unmount storage=<storage-name>  
unmount [ jobId=<id> | job=<job-name> ]
```

Stop

```
help jobId=<number-list> job=<job-name> ujobid=<unique-jobid> all
```

Diferentemente do comando **cancel**, o **stop** interrompe imediatamente os trabalhos selecionados, mas permite que eles sejam retomados posteriormente (com o comando **resume**). Se arquivos já foram copiados no backup, ele continua a partir do ponto que foi interrompido.

```
*stop  
Select Job(s):  
1: JobId=333 Job=BkOffice.2016-02-29_15.51.46_12  
2: JobId=334 Job=BkpOffice.2016-02-29_15.51.53_13  
Choose Job list to stop (1-2): 1  
JobId=333 Job=BkpLibreOffice.2016-02-29_15.51.46_12  
Confirm stop of 1 Job (yes/no): yes  
2001 Job "BkpOffice.2016-02-29_15.51.46_12" marked to be stopped.
```

Update

Este comando é **absolutamente necessário** em duas situações:

1. Se você já tiver feito backups, mas quiser alterar retenções, duração de uso de volume ou qualquer outro aspecto do volume nas configurações de pools, você precisa executar **"update volumes fromallpools"** após alterar a configuração, se você deseja que as

mudanças afetem os volumes criados anteriormente.

2. Depois de remover ou inserir quaisquer fitas na biblioteca de fitas, você deve fazer um **update slots** para que o Bacula possa estar ciente das alterações. Informações sobre slots e se cada fita está no robô (ou não) são escritas no catálogo do Bacula. Se sua biblioteca de fitas não possuir um leitor de código de barras, o comando para a fita de inventário é um pouco diferente e mais lento: **update slots scan**. Montará cada fita na movimentação para descobrir seu nome.

Você também pode usar a atualização para alterar qualquer aspecto do volume: a **Pool** a qual pertence; seu **status** (por exemplo: de used para archive se você não quiser que ele seja sempre reciclado) etc. Interativamente, faça: **update > Volume Parameters** para mostrar tudo que pode ser alterado:

```
Volume Status
Volume Retention Period
Volume Use Duration
Maximum Volume Jobs
Maximum Volume Files
Maximum Volume Bytes
Recycle Flag
Slot
InChanger Flag
Pool
Volume Files
Volume from Pool
All Volumes from Pool
```

Existem outros possíveis usos do comando update, mas alguns deles são menos importantes (por exemplo: o **update > Pool from resource** (configurações das pools é feito automaticamente quando o Director é reiniciado).

```
update
  stats
  snapshot
  pool=<poolname>
  slots storage=<storage> scan
    volume=<volname> volstatus=<status> volretention=<time-def> pool=<pool>
  recycle=<yes/no> slot=<number> inchanger=<yes/no> maxvolbytes=<size>
  maxvolfiles=<nb> maxvoljobs=<nb> enabled=<yes/no> recyclepool=<pool>
  actiononpurge=<action> allfrompool=<pool> fromallpools
```

Var

```
var <variable>
```

Este comando pega expressões e as expande, é basicamente usado para testar variáveis do Bacula. As variáveis devem ser utilizadas para fazer volumes rotulados automaticamente (geralmente para backup em disco) na diretiva de recurso de pool **LabelFormat**, mas também podem ser usadas para scripts Bacula. Por exemplo:

```
var $Pool
var $Day
```

O Apêndice 3 deste livro traz as variáveis possíveis atuais.

Version

`version`

Mostra a versão do Director.

Quit

`quit`

Sai do bconsole graciosamente (porque sair usando **Ctrl+C** é deselegante).

12.2 Comandos situacionais do Bacula

Alguns desses comandos são usados em situações muito específicas. Outros são usados muito raramente.

Add

`add [pool=<pool-name> storage=<storage> jobid=<JobId>]`

Mais simplório do que o comando `label`, adiciona volumes ao catálogo do Bacula e não tenta criar no disco ou na fita. Você pode usá-lo para criar vários volumes ao mesmo tempo, mas eles realmente não existem na mídia física até que você envie jobs de backup e o Bacula eventualmente os criar. É frequentemente usado para backups de disco, já que a rotulagem automática é mais fácil, mas às vezes pode ser usada também para criar volumes escritos por outro Bacula para permitir que o Bacula sincronize suas informações com uma nova instalação quando um backup é enviado para ele, mas há maneiras mais rápidas e seguras de fazê-lo.

Autodisplay

`autodisplay on/off`

Este comando pode ser usado para ativar a exibição automática de mensagens do Bacula no bconsole (algo como uma `tail -f`). As mensagens ainda só podem ser vistas uma vez e são impressas quando qualquer tecla é pressionada. É uma decisão muito pessoal usar esta funcionalidade, e ela afeta somente a CLI do bconsole.

Automount

`automount on/off`

Isso pode ser usado para desativar o recurso de montagem automática do Bacula ao receber um comando `label`. Normalmente, nunca há razão para desabilitar isso.

Llist

`llist`

O comando **llist** (lista longa) aceita as mesmas opções do **list**, mas traz todas as informações detalhadas do objeto selecionado do Bacula. Cada campo é impresso após uma quebra de linha e a saída é consideravelmente maior – mais usado para verificação de problemas. Apenas como exemplo, esta é a saída de um comando **llist pools**:

```
PoolId: 1 Name: Default
NumVols: 0
MaxVols: 0
UseOnce: 0
```



```
UseCatalog: 1
AcceptAnyVolume: 1
VolRetention: 1,296,000
VolUseDuration: 86,400
MaxVolJobs: 0
MaxVolBytes: 0
AutoPrune: 0
Recycle: 1
PoolType: backup
LabelFormat: *
PoolId: 2
Name: Recycle
NumVols: 0
...
```

Relabel

```
relabel
```

Esse comando é usado para corrigir um nome de volume já rotulado (por exemplo) quando ocorreu um erro de digitação. O comando completo é o seguinte:

```
relabel storage=<storage-name> oldvolume=<old-volume- name>
volume=<newvolume-name>
```

Observação 1: o status do volume **precisa** ser **purged** ou **recycled** se desejar reetiquetar volumes.

Observação 2: o Bacula precisa acessar o volume para mudar fisicamente seu rótulo. Por exemplo: se for uma fita, você precisa tê-la montada na unidade.

Setbandwidth

```
setbandwidth
```

Este novo comando do Bacula 7.x permite limitar dinamicamente o rendimento máximo de um job em execução ou futuros trabalhos de um cliente.

```
* setbandwidth limit=1000 jobid=10
```

*Observe que o valor especificado para o parâmetro de linha de comando limite está sempre em unidades de 1024 bytes (ou seja, o número é multiplicado por 1024 para dar o número de bytes por segundo). Como consequência, o limite acima de 1000 será interpretado como um limite de $1000 * 1024 = 1.024.000$ bytes por segundo.*⁹⁷

Uma maneira mais permanente de limitar a taxa de transferência para um trabalho específico ou vários trabalhos seria usar a diretiva de `MaximumBandwidth` nos recursos `job` ou `JobDefs` do `bacula-dir.conf`. Ex.: `MaximumBandwidth = 5Mb/s`.

Setdebug

```
setdebug
```

Pode ser usado para habilitar a depuração para cada um dos daemons do Bacula. Bastante útil para diagnosticar mau comportamento de algum dos daemons:

```
Ex.: setdebug level=nn [trace=0/1 client=<client-name>
| dir | director | storage=<storage-name> | all]
```

Normalmente, o nível usado para a investigação é 100. Se você definir “**trace = 1**”, o daemon especificado começará a ser executado no modo de rastreamento. Dessa forma, cada saída de depuração será redirecionada para um arquivo `bacula.trace`, no mesmo diretório dos daemons. Normalmente, o rastreamento é necessário somente para clientes Win32 onde a saída de depuração não pode ser gravada em um terminal ou redirecionada para um arquivo. Ao rastrear, cada mensagem de saída de depuração é anexada ao arquivo de rastreamento. Você deve excluir o arquivo explicitamente quando tiver terminado.

Outra maneira de ver as mensagens de depuração Bacula é apenas iniciar daemons com a opção **-d <nível>**, como sugerido antes na seção “Instalação do Bacula e resolução de problemas no início de daemons”.

Sqlquery

`sqlquery`

Este é um comando mais incomum, que permite que o usuário Bacula digite consultas SQL diretamente ao banco de dados Bacula. Cada linha inserida é concatenada à linha anterior até ser observado um ponto e vírgula (;). O ponto e vírgula encerra o comando, que é então passado diretamente para o mecanismo de banco de dados SQL. Quando a saída do mecanismo SQL é exibida, a formação de um novo comando SQL começa. Para encerrar o modo de consulta SQL e retornar ao prompt de comando do console, você insere um ponto (.) na coluna 1.

Uma maneira mais segura de executar consultas é testá-las e gravá-las previamente em um arquivo de texto e chamá-las usando o comando **query**.

Use

`use <database-name>`

Esta é uma situação muito rara e incomum: quando você configura um Director do Bacula para usar dois ou mais bancos de dados como catálogos (não é realmente necessário ou recomendado, uma vez que um banco de dados bem ajustado com recursos suficientes dificilmente será um gargalo para o Bacula). O comando `use` permite alternar entre esses catálogos.

Query

`query`

Este comando executa consultas de SQL escritas personalizadas a partir de um arquivo, chamado pela diretiva `QueryFile`, no recurso de diretório `bacula-dir.conf`. Bacula já traz muitas consultas de amostra, mas você pode criar novas apenas editando o arquivo de consulta. Se você construiu o seu Bacula a partir do código-fonte, você também pode precisar copiar a consulta de exemplo do local de origem para o diretório do Bacula. Por exemplo:

```
cp /usr/src/bacula-7.4.3/examples/sample-query.sql /etc/bacula/scripts/  
query.sql
```

Ao executar o comando de consulta, ele perguntará qual consulta você deseja executar:

Available queries:

- 1: List up to 20 places where a File is saved regardless of the directory
- 2: List where the most recent copies of a file are saved

```
3: List last 20 Full Backups for a Client
4: List all backups for a Client after a specified time
5: List all backups for a Client
6: List Volume Attributes for a selected Volume
7: List Volumes used by selected JobId
8: List Volumes to Restore All Files
9: List Pool Attributes for a selected Pool
10: List total files/bytes by Job
11: List total files/bytes by Volume
12: List Files for a selected JobId
13: List Jobs stored on a selected MediaId
14: List Jobs stored for a given Volume name
15: List Volumes Bacula thinks are in changer
16: List Volumes likely to need replacement from age or errors
17: List Volumes Bacula thinks are eligible for the changer
18: List Volumes by Volume:
19: List Volumes by Jobs:
20: List Volumes for a jobname:

Choose a query (1-20):
```

Opção **6** é útil para selecionar manualmente backups para restaurar de um cliente específico. A opção **16**, para detectar possíveis mídias desgastadas, provavelmente após vários anos, e substituí-las. **17** é útil se você precisar de dicas sobre as fitas usadas, mas já expiradas, que você deve trazer de volta para o robô, provavelmente candidatas a serem recicladas.

Wait

```
wait [jobid=nn] [jobuid=unique id] [job=job name]
```

*O comando wait faz com que o Director pause até que não haja trabalhos em execução. Este comando é útil em uma situação de lote, como testes de regressão onde você deseja iniciar um trabalho e esperar até que o trabalho seja concluído antes de continuar.*⁹⁸

Se especificado com um **JobId**, o comando **wait** aguardará que esse trabalho específico termine antes de continuar.

12.3 Comandos especiais de redirecionamento (@)

Os comandos de redirecionamento trazem muitas possibilidades de saída de resultados do bconsole para o sistema de arquivos, ou criação de arquivos que contenham scripts para serem lidos pelo bconsole. Isso permite a automação de qualquer processo:

```
@input <filename>
```

Executa no bconsole Bacula os comandos previamente escritos em um arquivo de texto (filename deve conter o caminho completo para o arquivo):

```
@output <filename> w/a
```

Alguns comandos bconsole podem ser maiores do que o limite de linhas configurado de seu terminal e talvez seja útil redirecioná-los. Este comando envia a saída dos comandos submetidos para um arquivo. Você pode escolher entre sobrescrever arquivo original (w) ou anexar (a). Se você não especificar qualquer arquivo, a saída só vai para o terminal. Por exemplo:

```
@output /tmp/bacula_report
list jobs
```

```
show filesets
version
@output
```

A saída dos três comandos ordinários será gravada em **/tmp/bacula_report**.

```
@tee <filename> w/a
```

Semelhante ao **@output**, mas também envia a saída para o terminal.

```
@sleep <seconds>
```

Um comando de **sleep** para scripts de Bacula.

```
@version
```

Exibe a versão atual do Bacula.

```
@# qualquer coisa
```

Comentários para scripts Bacula.

12.4 Redirecionamento do shell para o bconsole

Muitas tarefas do Bacula podem ser automatizadas usando shell script. Exemplo:

```
bconsole -c ./bconsole.conf <<END_OF_DATA
unmount storage=DDS-4
quit
END_OF_DATA
mt -f /dev/nst0 eject
```

Este exemplo de script foi chamado pelo último trabalho de backup do Bacula (RunAfterJob) para ejetar uma fita manual da unidade após o backup ter terminado. Mas este exemplo pode ser aplicado a vários outros propósitos.

Outra maneira fácil de redirecionar comandos para o bconsole, a partir do Shell:

- Executar um job de backup avulso já configurado:

```
echo "run job=BackupBaculaConfs level=full yes" | bconsole
```

- Monitorar a execução de um job de backup de um cliente, em tempo real, utilizando Shell:

```
watch -n 1 "echo status client=bacula-fd | bconsole"
```

A parte da saída importante que será atualizada a cada 1 segundo:

```
...
```

```
Running Jobs:
```

```
JobId 326 Job BackupHome.2015-08-21_12.00.10_26 is running.
```

```
Full Backup Job started: 21-Aug-15 12:00
```

```
Files=11,831 Bytes=1,384,850,011 AveBytes/sec=27,697,000 LastBytes/
sec=30,304,449 Errors=1
```

```
Bwlimit=0
```

```
Files: Examined=11,832 Backed up=11,831
```

```
Processing file: /home/hfaria/Documents/Python Developer'S Handbook.rar
```

```
...
```

E depois de um tempo...

```
...
```

```
Running Jobs:
```

```
JobId 326 Job BackupHome.2015-08-21_12.00.10_26 is running.
```

```
Full Backup Job started: 21-Aug-15 12:00
Files=13,839 Bytes=7,378,483,161 AveBytes/sec=38,429,599 LastBytes/
sec=42,558,371 Errors=1
Bwlimit=0
Files: Examined=13,840 Backed up=13,839
Processing file: /home/hfaria/Music/Bon Jovi - Love Ain't Nothing But A Four
Letter Word.mp3
...
```

12.5 Comandos automáticos do bconsole durante o backup (e substituição de caracteres)

Programar comandos do bconsole é outra maneira de automatizar qualquer operação no Bacula (por exemplo: um **update slots** para os robôs de fitas identificarem fitas trocadas nos slots se você esquecer de fazê-lo manualmente); ou aumentar o nível de informação nos logs (ex.: **show fileset=%f** para listar os diretórios que serão copiados naquele backup), uma vez que pode ser prático para a solução de problemas e talvez útil para enfrentar políticas de auditoria de segurança mais rígidas.

A definição do comando de script para o console é especificada dentro do recurso job (sub-recurso) para um trabalho específico; ou para todos os jobs no JobDefs.

- Exemplo de **update slots**, normalmente definido para uma primeira tarefa de backup de prioridade mais alta:

```
RunScript {
    RunsWhen = Before
    RunsOnClient = no
    Console = "update slots storage=storage_name"
}
```

- Exemplo de **show fileset=%f**, para ser exibido sempre no início das pastas e configurações do job de backup no FileSet (no JobDefs):

```
RunScript {
    RunsWhen = Before
    RunsOnClient = no
    Console = "show fileset=%f"
}
```

Observação: não funciona se o FileSet tiver espaços em seu nome.

A saída adicionada ao log seria semelhante a esta:

```
17-Set 17:27 hfaria-K46CB-dir JobId 767: shell command: run BeforeJob "/etc/
bacula/scripts/make_catalog_backup.pl MyCatalog"
17-Set 17:27 hfaria-K46CB-dir JobId 767: console command: run BeforeJob "show
fileset=Catalog"
17-Set 17:27 hfaria-K46CB-dir JobId 0: FileSet: name=Catalog
17-Set 17:27 hfaria-K46CB-dir JobId 0: O M
17-Set 17:27 hfaria-K46CB-dir JobId 0: N
17-Set 17:27 hfaria-K46CB-dir JobId 0: I /var/lib/bacula/bacula.sql
17-Set 17:27 hfaria-K46CB-dir JobId 0: N
```

O **%f (fileset)** é um dos caracteres de substituição Bacula, que pode ser usado como uma

variável na execução de qualquer script. Verifique a tabela completa:

%% = %

%b = Bytes do job

%c = Nome do client

%d = Nome do daemon (ex.: host-dir ou host-fd)

%D = Nome do Director (também válido para uso no FileDaemon)

%e = Status de saída do job

%f = FileSet do job (apenas para uso no lado do Director)

%F = Quantidade de arquivos do job

%h = Endereço do client

%i = JobId

%j = JobId único

%l = Job level

%n = Job name

%p = Nome da pool (apenas no lado do Director)

%s = A partir de determinado tempo

%t = Job type (backup, ...)

%v = Nome do volume (apenas no lado do Director)

%w = Nome do storage (apenas no lado do Director)

%x = Spooling habilitado? ("yes" ou "no")

⁹⁰ Diretiva "Label Format" em cada recurso Pool (bacula-dir.conf). O Storage Device (bacula-sd.conf) também deve ter a diretiva "Label media = yes" para que funcione (normalmente tem).

⁹¹ Com Bacula Versão: 7.4.0 (16 de janeiro de 2016).

⁹² Gravei um vídeo aprofundado sobre este assunto: <<http://www.bacula.com.br/analise-teorica-do-algoritmo-de-busca-de-volumes-do-bacula/>>.

⁹³ Disponível em: <http://www.bacula.org/7.4.x-manuals/en/console/Bacula_Console.html>.

⁹⁴ Disponível em: <<https://www.mail-archive.com/bacula-users@lists.sourceforge.net/msg22275.html>>.

⁹⁵ Tradução literal do texto disponível em: <http://www.bacula.org/7.4.x-manuals/en/console/Bacula_Console.html>.

⁹⁶ Esta seção do manual do Bacula está muito bem escrita, e eu traduzi literalmente com algumas notas entre colchetes: <http://www.bacula.org/7.4.x-manuals/en/main/Restore_Command.html>.

⁹⁷ Traduzido do manual do Bacula: <http://www.bacula.org/7.4.x-manuals/en/main/New_Features_in_7_0_0.html>.

⁹⁸ Fonte: <http://www.bacula.org/7.4.x-manuals/en/console/Bacula_Console.html>.

13

Restauração Interativa pela Linha de Comando

13.1 O menu do comando restore

O Bacula traz várias maneiras de fazer basicamente o mesmo: selecionar os jobs e arquivos a serem restaurados. No bconsole:

```
* restore
```

O **bconsole** pergunta como você deseja localizar o trabalho a ser restaurado:

Para selecionar o JobIds, você tem as seguintes opções:

- 1: Listar últimos 20 trabalhos executados
- 2: Listar trabalhos onde um determinado arquivo está salvo
- 3: Insira a lista de JobIds separados por vírgula para selecionar
- 4: Inserir uma lista com comandos SQL
- 5: Selecione o backup mais recente para um cliente
- 6: Selecione backup para um cliente antes de uma hora especificada
- 7: Insira uma lista de arquivos para restaurar
- 8: Insira uma lista de arquivos a restaurar antes de uma hora especificada
- 9: Encontre os JobIds do backup mais recente para um cliente
- 10: Encontre os JobIds para um backup para um cliente antes de uma hora especificada
- 11: Insira uma lista de diretórios a serem restaurados para os JobIds encontrados
- 12: Selecione restauração completa para uma data de trabalho especificada
- 13: cancelar

Selecionar item: (1-13):

As opções **2, 3, 5, 6 e 7** são definitivamente as mais importantes e vamos ficar com elas. As outras são mais situacionais.

A **opção 2** parece um comando “find” do shell. Basta inserir um nome de arquivo e o Bacula exibirá todos os jobs para um determinado cliente que contêm o arquivo. Muito útil quando o usuário não sabe com certeza o caminho para o arquivo, quando ele foi perdido ou qual versão ele deseja que seja restaurada.

Se você já sabe o **JobId** das tarefas que você deseja restaurar – por exemplo: um completo (full) e outro diferencial (**outro comando do bconsole fornece, o list jobs**) – você pode apenas **inserir-los entre vírgulas usando a opção 3**. Ao selecionar vários trabalhos, o Bacula irá fundi-los em uma única árvore de restauração, mantendo os arquivos mais atualizados entre eles. É a maneira mais básica de restauração. A listagem de jobs pode ser feita com o

comando **restore**, **opção 1**, mas apenas os últimos vinte trabalhos.

Outra maneira muito popular de restaurar é a **opção 5**, mas só se aplica se você precisar da última restauração do servidor. O Bacula seleciona entre os últimos backups completos e incrementais/diferenciais para construir sua árvore de restauração.

A **opção 6** é uma forma mais automática de selecionar trabalhos antes de um determinado ponto do tempo. O Bacula selecionará automaticamente os últimos backups completos e diferenciais; ou o último completo e incremental antes desse dado tempo.

A **opção 7** é outro método alternativo que pode ser útil se enfrentar grandes tempos de resposta ao criar a árvore de restauração para seleção. Ele permite selecionar arquivos para restaurar apenas digitando seu nome e caminho completo. Se não existir na base de backup, o Bacula avisará e nada será selecionado. Vários arquivos podem ser especificados e, em seguida, basta pressionar a tecla “Enter” em uma linha em branco para terminar a seleção e continuar com a restauração.

13.2 Árvore de seleção de arquivo para restauração da CLI

A maioria das opções anteriores do menu de restauração levará a uma ou mais dessas perguntas. Se solicitado a fornecer o **cliente**, trata-se da origem, ou seja, de onde o backup foi feito:

```
Defined Clients:
1: bacula-fd
2: hfaria-fd
Select the Client (1-2):
```

O FileSet também pode ser perguntado se há jobs de backup que foram configurados ou executados com dois conteúdos diferentes.

```
The defined FileSet resources are:
1: Catalog
2: Full Set
Select FileSet resource (1-2):
```

Depois de responder todas as perguntas, a árvore de arquivos para seleção será construída. As cruzes indicam uma barra de progresso baseada em texto e essa seleção pode demorar um pouco, de acordo com o número de arquivos que foram copiados neste trabalho (e outros direcionadores de desempenho do banco de dados):

```
You have selected the following JobIds:
43,55,56,59 Building directory tree for JobId 43 ... ++++++
+++++
Building directory tree for JobId 55 ... Building directory tree for JobId 56
... + Building directory tree for JobId 59 ...
4 Jobs, 342,907 files inserted into the tree.
```

Em seguida, o prompt de seleção de arquivos será mostrado (\$):

```
You are now entering file selection mode where you add
(mark) and remove (unmark) files to be restored. No files are initially
added, unless you used the “all” keyword on the command line.
Enter “done” to leave this mode.
```

```
cwd is: /
```


Os comandos dessa outra interface podem ser lembrados com o comando help:

Comando Descrição

=====

```
addadiciona dir / file para ser restaurado recursivamente, curingas permitidos
cdmuda o diretório atual
count conta arquivos marcados para restauração
delete exclui diretórios ou arquivos a serem restaurados, recursivamente
dir lista longa de arquivos do diretório corrente, curingas aceitos
done sair do modo de seleção
estimate estimar tamanho do restore
exit o mesmo que o comando done
find encontrar arquivos, curingas aceitos
help imprime essa ajuda
ls lista o diretório atual, curingas aceitos
lsmark lista os arquivos marcados para restauração abaixo do diretório atual
mark marca arquivos ou diretórios para restaurar, recursivamente
markdir marca apenas o diretório para restauração (sem arquivos)
pwd imprimir o diretório atual
unmark desmarca diretório ou arquivo recursivamente
unmarkdir desmarca apenas um diretório da restauração, não recursivo
quit sair sem restaurar
? imprimir ajuda
```

Se você precisa de um único arquivo ou diretório, você pode subir até eles com o comando **cd** e listar o conteúdo das pastas com os comandos **ls** ou **dir**. Você pode selecionar arquivos ou pastas recursivamente com o comando “**mark nome_arquivo**”. E se você precisar restaurar todos os arquivos dos trabalhos selecionados, pode fazer:

```
$ mark *
346,620 arquivos marcados.
```

Depois de selecionar todos os arquivos e pastas para restaurar, termine a operação com o comando **done**. Você está agora a apenas um passo de submeter a tarefa de restauração:

```
Run Restore job
FebNameFiles
$var$baepla/working/dhcp-dir.restore.1.bsr
$map$bacula-restores
$map$se:
$map$Set:
$map$upClient:
$map$ofClient:
$map$ggr:
When: 05-21 16:23:05
MyCatalog
Priority:
OK to run? (yes/mod/no):
```

Essas são as opções mais importantes a considerar:

- **Where (onde):** em que diretório do cliente de restore os arquivos serão restaurados. Se inexistente, o Bacula criará.
- **Replace (substituir):** o que fazer se houver arquivos com o mesmo nome no local que

you are restoring.

- Backup Client (cliente de backup):** qual cliente é o proprietário dos arquivos que você está restaurando? Normalmente, nunca mudar isso.

- Restore Client (cliente de restore):** em que cliente do Bacula os arquivos serão restaurados? Isso permite a restauração cruzada, bastante útil quando você não pode restaurar para a mesma máquina proprietária dos arquivos originais.

After the appropriate changes, type **yes** (sim) to send the **Job**. Check the messages in the log.

Observação: nunca use o comando “run” para criar jobs de restore, a menos que você saiba o que está fazendo.

14

Recuperação de Desastres do Servidor Bacula

14.1 Restauração do catálogo

14.1.1 A partir de um dump anteriormente criado

MySQL

- Na pasta que você tem o arquivo de dump, entre no console do **MySQL**:
`mysql`
- Crie um banco de dados vazio para restaurar:
`create database bacula`
- Alterne para o novo banco:
`use bacula`
- Execute a importação:
`\. bacula.sql`
- Você também poderia ter feito isso, alternativamente, a partir do shell:
`mysql bacula < bacula.sql`

Observação: se você estiver lidando com sites remotos, você pode usar **nohup** para lidar com erros de importação e desconexão longos:

```
nohup mysql bacula < bacula.sql &
```

PostgreSQL

```
su postgres  
psql bacula -f bacula.sql
```

Exercício: a partir de um ambiente de testes do Bacula, gere o **dump**, faça um **drop** no banco de dados e importe-o novamente.[99](#)

14.2 bscan

Por alguma razão, suas retenções de volumes ou de informações no catálogo não foram bem configuradas e você perdeu informações sobre o volume no catálogo do Bacula, ou talvez apenas a máquina do Bacula Director foi afetada por um desastre. A ferramenta bscan

permite ler o volume e restaurar informações sobre ele novamente no catálogo do Bacula. Depois disso, você será capaz de (por exemplo): restaurar um único arquivo, listar os trabalhos que foram armazenados nesses volumes etc.

É aconselhável parar os daemons do Bacula para executar o `bscan`. Ele deve ser executado sob o shell e este é um exemplo em fita:

```
bscan -s -m -v -V Daily-0 /dev/st0 -u root -P passwd  
[command] [volumes_names] [tape_device] [db_credentials]
```

Para volumes em disco:

```
bscan -s -m -v -V Daily-0 /mnt/backups -u root -P passwd  
[command] [volumes_names] [disk_device] [db_credentials]
```

Observação 1: você também pode pesquisar vários volumes com o separador | (pipe):

```
-V "Daily-0|Daily-1"
```

Observação 2: se você estiver usando outro método de autenticação de banco de dados, talvez não seja necessário especificar credenciais de acesso ao banco.

No final de um `bscan` ele irá imprimir um sumário de cada informação restaurada (arquivos, volumes, pools etc).

Exercício: exclua um volume com dados de backup de teste do seu catálogo do Bacula (`bconsole`, comando `delete`) e restaure suas informações com `bscan`.

14.3 bls e bextract

Diferentemente do `bscan`, o **bextract** é usado para extrair arquivos de um volume do Bacula. Mas antes de fazer isso você pode querer listar todo o conteúdo do volume, o que pode ser feito com **bls**.

```
bls -V nome-volume /dev/nst0
```

Observação: se estiver usando volumes baseados em disco, basta alterar o dispositivo da unidade de fita com o valor do **Archive Device** (`bacula-sd.conf`, que é o diretório onde estão os volumes).

Você deve ter notado que não precisa de um job do Bacula ou do catálogo para usar o **bextract**, mas você pode ter menos controle ou granularidade dos arquivos que você precisa restaurar. Existem muitas opções para fazer isso:

```
bextract -i lista-inclusão -V nome-volume /dev/nst0 /tmp
```

Com a opção `-i`, o `bextract` vai restaurar apenas os arquivos que correspondem ao nome completo do arquivo, especificado em uma lista (que suporta também expressões regulares) e que você deverá criar na forma de um arquivo texto. Você também pode criar uma lista de exclusão com uma opção `-e`, se você quiser fazer o caminho oposto.

Um exemplo de lista de inclusão para restaurar os arquivos de configuração do Bacula no servidor e dump do banco de dados em qualquer diretório que tenha sido gerado:

```
*bacula.sql  
/etc/bacula/*.conf
```

No entanto, você pode ter várias versões do mesmo job de backup escritas no mesmo volume. Por causa disso, você pode ainda manipular o arquivo de **bootstrap** Bacula – que é um pequeno índice de texto dos trabalhos sem volume, para restaurar dados de um volume específico. Você pode editar o bootstrap criado anteriormente para selecionar o trabalho que você precisa restaurar e usá-lo com `bextract`:

```
bextract -b arquivo-bootstrap -V nome-volume /dev/nst0 /tmp
```

⁹⁹ O script de backup do catálogo do Bacula “Run Before Job” pode ser usado para gerar o dump, como o Bacula normalmente faz para efetuar o backup do banco de dados.

15

Atualizando o Bacula

Este capítulo deve ser usado para atualizar qualquer Bacula – não apenas da versão 5.x para 7.x. No entanto, se você tiver uma versão mais antiga, como 2.x, você deve atualizar para 3.x e executar o script (por exemplo: `/etc/bacula/scripts/update_bacula_tables`) para atualizar para uma versão 5.x. Novamente, execute o script da nova instalação com o próprio nome para finalmente atualizar uma estrutura do banco de dados.

Você deve atualizar o **Director** e os **Storage Daemons** ao mesmo tempo e sempre para uma versão comum. No entanto, os clientes podem ser atualizados gradualmente. Em outras palavras: as versões do **Director** e storage devem ser iguais e nunca inferiores às versões do **File Daemon**.

Os arquivos de configuração das versões instaladas e mais antigas são sempre mantidos para implantar novas versões mais atualizadas do Bacula.

15.1 Etapas gerais de atualização

- Faça backup simples dos arquivos de configuração `/etc/bacula`. Por exemplo, como root:

```
mkdir /root/updatebkp
cp -r /etc/bacula /root/updatebkp/
```

- Faça um dump do banco de dados do Bacula também para recuperação de possíveis desastres na atualização. Por exemplo:

```
mysqldump bacula -u bacula -p[mysql_password] > /root/updatebkp/
bacula.sql
```

- Siga os procedimentos de **instalação por pacotes ou instruções para compilação dos binários** a partir do Capítulo 6, **Instalação do Bacula**.

Dica: se atualizando um Bacula anterior construído a partir da instalação de origem (compilação), seria aconselhável usar as mesmas opções utilizadas no comando `./configure` anterior, para garantir que todos os arquivos binários ou Bacula corresponderão e atualizarão uma instalação antiga. Isso pode ser recuperado a partir do **config.log**, sob o caminho do código-fonte anterior (e isso é mais uma razão pela qual você não deve usar `/tmp` para baixar e compilar seu código, mas um local padrão, como `/usr/src`)¹⁰⁰. Por exemplo:

```
cat /usr/src/bacula-7.4.1/config.log |grep "$ ./configure"
```

Saída:

```
$ ./configure --with-readline=/usr/include/readline --disable-conio --
```

```
bindir=/usr/bin --sbindir=/usr/sbin --with-scriptdir=/etc/bacula/scripts --  
with-working-dir=/var/lib/bacula --with-logdir=/var/log --enable-smartalloc  
--with-mysql
```

- Atualize sua estrutura do banco de dados do Bacula se necessário (**2.x para 3.x, 3.x para 5.0, 5.0 para 5.2**: aqueles onde os lançamentos já exigiram essa atualização). Este script de banco de dados pode variar de acordo com sua pasta de scripts do Bacula e o método de autenticação do banco de dados.

```
/etc/bacula/scripts/update_bacula_tables
```

- Reinicie os daemons Bacula.

- Teste tudo. Use o comando **status** e a **estimate listing** para testar o estado dos seus clientes, storage e jobs. Execute um job de backup de teste.

¹⁰⁰ É possível também desinstalar completamente uma instalação por compilação do Bacula executando um comando **make uninstall**, também a partir do diretório de extração de código-fonte antigo. Observe que isso provavelmente também excluirá todos os arquivos de configuração do Bacula dentro do **/etc**.

Backup de Aplicações Específicas

16.1 O problema

Muitas aplicações são difíceis de restaurar, ou é simplesmente impossível de fazer isso apenas copiando e recuperando arquivos e conteúdo da pasta. Normalmente, estes são o backup do banco de dados e o de nível de imagem de máquinas virtuais, mas há vários outros exemplos. Mesmo os sistemas operacionais podem exigir procedimentos especiais: o Microsoft Windows requer um “dump” especial de seu registro e outros arquivos de sistema abertos que só pode ser feito com a execução de um backup nativo do systemstate.

O procedimento especial para cada aplicação só pode ser conhecido pela leitura de sua documentação, e às vezes pode haver mais de uma opção. Adiante existem vários exemplos, mas seria impossível cobrir todas as aplicações e versões diferentes. Mesmo aplicativos do mesmo tipo, como bancos de dados (por exemplo: Oracle, MySQL, PostgreSQL) podem ter diferentes maneiras de protegê-los com backup.

Existem várias maneiras de automatizar e executar essas rotinas de backup intermediário das aplicações. Da pior (em geral) para a melhor:

a) **Programar uma execução de script com um agendador externo (por exemplo, cron do Linux)** que faça o dump dos dados do aplicativo para um diretório e então copiar essa pasta com o Bacula através de um job de backup (incluindo o FileSet). **Esse método é considerado inseguro**, pois o script pode não ser executado e o administrador de backup pode não estar ciente disso. Além disso, se a execução do script demorar mais do que o normal, o trabalho de backup pode não copiar as informações esperadas, e o inverso pode geralmente acontecer. Outra desvantagem é a necessidade de usar espaço em disco para armazenar esse dump temporário.

b) **Criar um script antes do trabalho para parar o aplicativo e então copiar o conteúdo das pastas durante o backup.** E um script DEPOIS do job de backup deve supostamente iniciá-lo novamente. A maioria dos softwares de backup tem a capacidade de executar scripts em todos os sistemas operacionais, e eles podem ser usados para isso. **Entretanto**, parar aplicativos nem sempre é possível devido ao tempo de inatividade consequente, ou **a recuperação da aplicação através da simples cópia do diretório pode ser difícil** (por exemplo: restaurar uma única tabela de um banco de dados de um serviço com vários bancos).

c) **Usar um script antes do job de backup (ClientRunBeforeJob) para salvar esse dump de aplicativo em qualquer pasta e, em seguida, copiá-la pelo Bacula através do mesmo job.** Este método é semelhante, mas um pouco melhor do que o item “a”, uma

vez que agora o script de dump e o backup são sincronizados. Além disso, é possível controlar a execução do script (por exemplo: abortar o job de backup no caso de o script terminar com erro etc.). A necessidade de espaço em disco temporário para armazenar o dump pode ser uma desvantagem, e um outro script pós job de backup (**ClientRunAfterJob**) pode ser utilizado para remover o dump do disco local da máquina cliente.

d) **Em vez de salvar dumps de aplicativos para o disco do cliente, enviá-los para um FIFO (também chamado de pipe nomeado).** Esta é uma ótima maneira de fazer backup de aplicativos, uma vez que pode atender a praticamente todos. Em vez de salvar dumps diretamente em fitas (exemplo), eles devem ser direcionados para um pipe e, em seguida, o software de backup vai ler os dados diretamente desse pseudoarquivo. Não há uso de disco adicional, pois os dados vão diretamente do aplicativo para o backup através de um fluxo (stream).¹⁰¹ O backup do FIFO pode ser feito usando scripts, mas pode ser complicado, já que cada pipe deve ter um nome exclusivo, ser criado antes do job de backup e removido depois. Além disso, cada FIFO só pode receber fluxo de dados (dump por dump, arquivo por arquivo): não é possível enviar vários diretórios ou arquivos para um FIFO, a menos que o tar ou soluções similares sejam usados para empacotá-los. Felizmente, tanto o **Bacula Community** quanto o **Enterprise** possuem um plug-in genérico para automatizar isso: o **bpipe**. Ele gerencia automaticamente a escrita e a leitura da criação do FIFO, sendo necessário apenas especificar quais comandos farão o trabalho. Ao restaurar (por exemplo), é possível usar um comando para restaurar dados diretamente para o aplicativo (por exemplo: importando um banco de dados ou a máquina virtual para o hypervisor), também contando com menos demanda de espaço em disco para restauração. Este recurso é detalhado na próxima seção.

e) **Usar plug-ins de backup específicos para cada aplicação.** É preciso muito esforço para desenvolver plug-ins de backup para a variedade de aplicações de mercado e suas diferentes versões. É por isso que geralmente só podem ser encontrados em software de backup proprietário (por exemplo: **Bacula Enterprise**). **Centros de dados médios e grandes** precisam necessariamente ter esses plug-ins, uma vez que eles são mais automatizados, facilmente gerenciados e possuem outros recursos avançados, como backup diferencial em nível de blocos de dados, restauração granular (por exemplo: uma única tabela de um banco de dados), implantação automatizada e restauração de aplicações automatizadas, deduplicação global (entre outros). Esses plug-ins realmente melhoram a segurança do backup, reduzem drasticamente as horas de trabalho humanas e os custos de backup da demanda por storage. Todos os benefícios geralmente compensam o investimento na compra de licenças de software de backup (**especialmente aquelas que não possuem taxas proporcionais ao tamanho dos backups**).

16.2 Instalação e visão geral do plug-in Bacula bpipe

Esta é a maneira mais rápida, mais fácil, mais automatizada e econômica de fazer backups quentes de aplicativos específicos, tais como PostgreSQL, MySQL, Firebird, Oracle, bancos de dados LDAP; exportação dos snapshots de máquinas virtuais Xen e KVM; pacotes “tar” com vários arquivos; e mais.

O plug-in é executado no lado File Daemon (cliente) e é fornecido no diretório `/src/plugins/fd/bpipe-fd.c` do código-fonte do Bacula. Quando o plug-in é compilado e vinculado ao objeto compartilhado dinâmico resultante (DSO), ele terá o nome `bpipe-fd.so` e ficará no diretório de plug-ins do Bacula.

16.2.1 Habilitar o bpipe

Configurando o diretório de plug-ins do File Daemon

Edite o arquivo de configuração (`bacula-fd.conf`) do File Daemon e certifique-se de que o caminho definido no Plugin Directory aponte para a pasta onde o `bpipe-fd.so` foi instalado. Normalmente, é suficiente apenas descomentar a linha (como está). Exemplo:

```
Plugin Directory = /usr/lib64
```

Usuário de execução do bacula-fd

Devido à execução de comandos diferentes do sistema operacional, geralmente é aconselhável executar `bpipe` File Daemon como `root`. Isso pode ser feito modificando o valor das variáveis `BUSER` e `BGROUP` no script de início do Bacula (por exemplo: `/etc/init.d/bacula-fd start`).

```
BUSER=root
```

```
BGROUP=root
```

Configurando o uso do bpipe no recurso FileSet (bacula-dir.conf)

O plug-in `bpipe` é especificado na seção `Include` do recurso `FileSet` do seu job. Esta é a sintaxe completa da diretiva de Plugin conforme interpretada pelo `bpipe`, onde dois pontos (`:`) é o separador dos campos, e **mais de um bpipe pode ser utilizado no mesmo Include, em linhas diferentes**:

```
Plugin = "campo1:campo2:campo3:campo4"
```

Onde:

`campo1` é o nome do plug-in com sufixo `-fd.so` removido. Nesse caso:

```
Plugin = "bpipe:campo2:campo3:campo4"
```

`campo2` especifica o caminho completo exclusivo do nome de arquivo¹⁰² para o **named pipe** que será usado para receber dados. Observe que também é esse nome de arquivo que aparecerá no conteúdo das listas de backup e que será visto pelo usuário na árvore de arquivos de restauração. Por exemplo, se o valor for `/fifo/anydb.sql`, os dados copiados pelo plug-in serão colocados sob esse caminho e nome de arquivo (**a pasta precisa existir**). Escolha uma convenção de nomenclatura que seja única para cada pipe, a fim de evitar duplicidade com um caminho e nome de arquivo que realmente existe no seu sistema **ou entre os pipes**.

```
Plugin = "bpipe:/fifo/anydb.sql:campo3:campo4"
```

`campo3` para o plug-in `bpipe` especifica o comando que irá gerar os dados do dump para o FIFO durante o backup¹⁰³. A maioria dos aplicativos tem opções para enviar dados para a saída padrão do sistema operacional (por exemplo: **mysqldump** sem qualquer redirecionamento¹⁰⁴; **tar** com um traço (-) em vez de um nome de arquivo etc.), e essa é a informação que vai para o FIFO. O `/dev/stdout` também pode ser usado em vez de nomes de arquivo em comandos que exigem um **arquivo de saída**. Observe que **nunca é possível** (ou

realmente necessário) usar o redirecionamento de shell (< ou >) nessa configuração: não é bem interpretado pelo Bacula. **Por exemplo:**

```
Plugin = "bpipe:/fifo/anydb.sql:mysqldump -u root -p123456 --databases  
anydb:campo4"
```

campo4 para o **bpipe** especifica o comando que é chamado pelo plug-in durante uma futura restauração, a fim de gravar os dados **FIFO** de volta para o sistema.

Aqui existem dois métodos principais básicos:

- Definir um comando para **restaurar o dump para um arquivo real no sistema de arquivos** (em seguida, manipulá-lo e restaurar para o aplicativo, fora do Bacula).

```
Plugin = "bpipe:/fifo/anydb.sql:mysqldump -u root -p123456 -databases  
anydb:dd of=/root/anydb.sql"
```

- Configurar um comando para restaurar dados **diretamente do Bacula para o aplicativo**, com suas respectivas opções de importação ou restauração.

```
Plugin = "bpipe:/fifo/anydb.sql:mysqldump -u root -p123456 --databases  
anydb:mysql -u root -p123456"
```

Em qualquer caso, a opção de caminho de restauração do Bacula (Where) é ignorada ao restaurar os pipes, e os dados são exatamente o que foi configurado na configuração do bpipe.

Exemplos ilustrativos

Um **echo**, restaurando para um arquivo:

```
Plugin = "bpipe:/fifo/test.txt:/bin/echo fifo_teste:dd of=/tmp/test.txt"
```

Outro exemplo de backup usando tar (/etc), descompactando em /tmp automaticamente durante a restauração:

```
Plugin = "bpipe:/fifo/test.tar:/bin/tar -cf - /etc:/bin/tar -C /tmp -xvf -"
```

FileSet com bpipe através de includes dinamicamente gerados

Escrever cada configuração de bpipe pensando em restauração granular (por exemplo: um FIFO para cada banco de dados ou exportação de máquina virtual) pode ser difícil e é um processo muito manual.

No entanto, é possível automatizar a configuração Bacula FileSet para gerar automaticamente a diretiva de plug-in do bpipe, a fim de fazer o backup de todos os bancos de dados de um serviço MySQL (por exemplo). Isso também pode ser feito chamando um script shell:

```
FileSet {  
  Name = "Full Set"  
  Include {  
    Options {  
      signature = MD5  
    }  
    Plugin = "|/etc/bacula/scripts/mysql_bpipe.sh"  
  }  
}
```

E o script deve fornecer uma lista de todos os nomes de bancos de dados MySQL que serão

copiados:

```
#!/bin/bash
```

```
# Local sugerido: /etc/bacula/scripts/mysql_bpipe.sh
```

Substitua esse valor pela senha do usuário root do MySQL ou use um método de autenticação mais seguro.

```
PSW=123456
```

```
for DBNAME in $(echo "show databases;" | mysql -u root -p$PSW | tail -n+2 |  
grep -v _schema)  
do  
echo "bpipe:/fifo/$DBNAME.sql:mysqldump -u root -p$PSW --databases  
$DBNAME:mysql -u root -p$PSW"  
done
```

Apenas por curiosidade, esta é uma saída de amostra deste script. Ao ler este, o Bacula vai pensar que são quatro diferentes linhas de `Plugin = "..."`:

```
bpipe:/fifo/bacula.sql:mysqldump -u root -p123456 --databases bacula:mysql -u  
root -p123456  
bpipe:/fifo/mysql.sql:mysqldump -u root -p123456 --databases mysql:mysql -u  
root -p123456  
bpipe:/fifo/mythconverg.sql:mysqldump -u root -p123456 --databases  
mythconverg:mysql -u root -p123456  
bpipe:/fifo/sys.sql:mysqldump -u root -p123456 --databases sys:mysql -u root  
-p123456
```

Tenha em mente que este é apenas um exemplo e outros arranjos podem ser feitos. Ex.: **excluindo um dado banco de dados** (`| grep -v nome_dos_bancos` a serem excluídos do backup), ou mesmo criando FIFO a partir de algumas tabelas que são mais críticas e de restauração mais frequentemente solicitada (isso é feito especificando o nome da tabela após os bancos de dados no comando `mysqldump`...).

Observação: a melhor maneira de solucionar problemas de backup e restauração do bpipe está em executar o **bacula-fd** no **modo de depuração** (pare o daemon e inicie com a opção `-d`; ex.: **bacula-fd -d 200**). Esta é a única maneira de obter informações completas da execução de comandos do leitor e do gravador.

16.3 Exemplos de backup de aplicações com bpipe

16.3.1 Bancos de dados

MySQL

Isso foi abordado na última seção como um exemplo geral bpipe: volte as páginas para verificar.

PostgreSQL

Este método é muito semelhante ao usado para o MySQL. A única diferença é que, para importar automaticamente o banco de dados restaurado para o serviço PostgreSQL, o dump deve ser gerado no modo binário.

- Restaurando para um arquivo:

```
Plugin = "bpipe:/fifo/anydb.sql:sudo -u postgres pg_dump heitorpg:dd of=/root/anydb.sql"
```

•Restauração direta para o PostgreSQL (mudança de comandos de leitura e escrita do FIFO):

```
Plugin = "bpipe:/fifo/anydb.sql:sudo -u postgres pg_dump --format=c heitorpg:sudo -u postgres pg_restore -d template1 -C"
```

•Criação dinâmica de texto de backup de configuração para backup granular de todos os bancos através de script (exemplo)¹⁰⁵:

```
#!/bin/sh
# Nome sugerido: \\|/etc/bacula/scripts/bpipe-pg-db
# Define o comando de acesso ao postgresql
SUCMD="su - postgres -c"

# Pega o nome dos bancos de dados
for dbname in ` $SUCMD 'psql -d template1 -q -U postgres -t' <<EOF
select datname from pg_database where not datname in
('bacula','template0') order by datname;
EOF
~
do
echo "bpipe:/fifo/$dbname.dump:$SUCMD '/usr/bin/pg_dump --format=c $dbname':$SUCMD 'pg_restore -d template1 -C'"
done
```

FileSet exemplo para execução do script:

```
FileSet {
  Name = "PG Database Fileset"
  Include {
    Plugin = "\\|/etc/bacula/scripts/bpipe-pg-db"
  }
}
```

Oracle

Este é um banco de dados que eu não tive a oportunidade de testar muitas vezes, mas este comando pode funcionar com FIFO (teste antes de usar):

```
Pipe = "bpipe:/fifo/database.dmp:su - oracle exp database/password file=/dev/stdout log=/var/database.log:dd of=/root/database.dmp"
```

Firebird

Firebird nasceu pronto para bpipe: tem suas próprias palavras-chave para ler ou escrever para FIFO. Este é um exemplo onde a restauração será feita automaticamente para o Firebird:

```
Plugin = "bpipe:/fifo/anydb.sql:gbak -backup anydb stdout:gbak -replace stdin anydb_2"
```

O nome distinto (anydb_2) no comando de restauração é para não substituir o banco de dados antigo neste momento, mas ele pode ser alterado a critério do administrador.

LDAP

•Restaurando dados de um arquivo:

```
Plugin = "bpipe:/fifo/ldap.ldif:slapcat -l /dev/stdout:dd of=/root/ldap_restore.ldif"
```

•Restauração para LDAP:

```
Plugin = "bpipe:/mnt/fifo/ldap.ldif:slapcat -l /dev/stdout:/usr/sbin/
```

```
slapadd -l /dev/stdin"
```

Observação: sempre pare o daemon LDAP antes de restaurar com o comando slapadd.

• Outro exemplo, com filtragem de domínios para o dump:

```
Plugin = "bpipe:/tmp/ldap_bkp.ldif:slapcat -f /etc/ldap/slapd.conf -b  
"dc=ifnmg,dc=edu,dc=br" -l /dev/stdout:tee /tmp/ldap_restore.ldif"
```

16.3.2 Imagem de máquinas virtuais

As máquinas virtuais de nível de imagem podem ser muito mais complicadas. Geralmente, é necessário iniciar um snapshot sobre ela e exportá-la depois de terminado. Essa é a informação desejada para armazenar no Bacula.

Citrix Xen

• Método mais seguro de restauração de cada máquina para arquivo (substitua **vm_name**, **vm_uuid** e **vm_name_snps** com o seguinte: nome da VM para backup, UUID dessa VM e um nome significativo exclusivo para o snapshot, respectivamente):

```
Plugin = "bpipe:/fifo/vm_name.xva:snapshot=`xe vm-snapshot uuid=vm_uuid  
new-name-label=vm_name_snps` && xe template-param-set is-a-  
template=false uuid=$snapshot && xe vm-export vm=$snapshot filename=:dd  
of=/root/vm_name.xva"
```

• Um a um, restaurando automaticamente o Xen Server (também é necessário definir o **xen_storage_uuid**, que é o usado para restaurar a VM):

```
Plugin = "bpipe:/tmp/vm_name.xva:snapshot=`xe vm-snapshot uuid=vm_uuid  
new-name-label=vm_name_snps` && xe template-param-set is-a-  
template=false uuid=$snapshot && xe vm-export vm=$snapshot filename=:xe  
vm-import filename=/dev/stdin sr-uuid=xen_storage_uuid"
```

Observação: para buscar o uuid de armazenamento, digite no shell: **xe pbd-list**.

• Configuração dinâmica de todos os backups VM bpipe (**restauração de imagem para arquivo**):

```
#!/bin/bash  
#  
# /etc/bacula/scripts/bpipe_xen.sh  
#  
# Script para gerar Bacula FileSet bpipe sintaxe configuração para  
backup  
# Todos com Citrix Xen VM.  
#  
# Autoria: Heitor Faria (Copyleft: todos os direitos invertidos).  
# Testado por: Elias Pereira  
#  
# Backups Citrix Xen instalado no Bacula Cliente, como este (por  
exemplo):  
#  
# Plugin = "\\|/etc/bacula/scripts/bpipe_xen.sh"  
#  
for UUID in $(xe vm-list power-state=running is-control-domain=false |  
grep uuid | cut -d: -f2 - | tr -d \ )
```

```
do
VM_NAME=$(xe vm-param-list uuid=$UUID | grep -i name-label | cut -d: -f2- | tr -d \ )
SNAP_UUID=`xe vm-snapshot uuid=$UUID new-name-label=$VM_NAME.snps`
xe template-param-set is-a-template=false uuid=$SNAP_UUID
echo "bpipe:/var/$VM_NAME.xva:xe vm-export vm=$SNAP_UUID filename=:dd
of=/mnt/$VM_NAME.xva"
done
```

•**Importante consideração para todos os métodos anteriores:** é aconselhável remover snapshots gerados manualmente após o backup, ou de vez em quando. Por exemplo:

```
xe vm-uninstall force=true vm=nome_vm_snps
```

É possível automatizar esse exemplo de script usando o **ClientRunAfterJob** para excluir todos os snapshots anteriores do Xen, já que o job do Bacula já fez o backup das máquinas virtuais:

```
#!/bin/bash
#
# /etc/bacula/scripts/after_job_xen_snapshots_delete.sh
#
for i in `xe snapshot-list --minimal | sed -e 's/,/\ /g` ;
do
    xe snapshot-uninstall force=true uuid=$i ;
done
```

Vmware

Pode ser doloroso realizar o backup de máquinas virtuais no sistema ESXi sem plug-ins específicos, principalmente porque seu desenvolvedor bloqueia a instalação de basicamente qualquer pacote nas versões mais recentes, incluindo o Bacula File Daemon.

Existe uma solução alternativa, no entanto: há um script de código aberto chamado **GhettoVCB**¹⁰⁶ que automatiza o backup e a restauração de nível de imagem das máquinas virtuais. É implementar e ver a configuração que é explicada no meu site:

<<http://www.bacula.com.br/backup-maquinas-virtuais-vmware-com-bacula-e-gettovcb/>>

O script salva as máquinas virtuais exportadas localmente no sistema de arquivos ESXi e sugere que podem ser movidas para um armazenamento de backup usando compartilhamentos NFS (que é um protocolo muito lento para isso).

Dito isso, há um exemplo de configuração de job Bacula para uma única máquina virtual a seguir:

•Programar um script ClientRunBeforeJob para executar o backup de uma única VM usando:

```
ClientRunBeforeJob = "sshpass -p 'esxi_passwd' ssh -o
StrictHostKeyChecking=no root@192.168.10.55 /Backup/ghettoVCB/
ghettoVCB.sh -g /Backup/ghettoVCB/ghettoVCB.conf -m VM_NAME"
```

•Configure a cópia da pasta criada no ESXi para a exportação da VM para o storage do Bacula usando FIFO, pois não é possível instalar um File Daemon lá. Exemplo de configuração do bpipe no include do FileSet:

```
Plugin = "bpipe:/fifo/VM_NAME:sshpass -p 'esxi_passwd' ssh -o
StrictHostKeyChecking=no root@192.168.10.55 /bin/tar -c /vmfs/volumes/
NFS01/233/VM_NAME:sshpass -p 'esxi_passwd' ssh -o
StrictHostKeyChecking=no root@192.168.10.55 /bin/tar -C /vmfs/volumes/
```

```
NFS01/233/ -xvf -"
```

- Um script pós job de backup (ClientRunAfterJob) para apagar a pasta criada pelo GhettoVCB, já que ela já está copiada:

```
ClientRunAfterJob = "sshpass -p 'esxi_passwd' ssh -o  
StrictHostKeyChecking=no root@192.168.10.55 rm -rf /vmfs/volumes/  
NFS01/233/VM_NAME"
```

Outra possibilidade é usar o backup GhettoVCB diretamente para um FIFO, mas isso exigiria a alteração do script. Esse recurso foi sugerido para o desenvolvedor, uma vez que eu não quis criar um fork do projeto.

16.3.3 Backup alternativo por FIFO usando shell script

Se por algum motivo não for possível o uso do plug-in bpipe, outra maneira de trabalhar com named pipes (FIFO) seria através de script shell no Linux. Exemplo:

- a)O FIFO é criado (ex.: no Linux, mkfifo /tmp/fifo).
- b)O conteúdo é enviado para um FIFO (ex.: echo "test" > /tmp/fifo), que sempre possui 0 bytes, mas aguardando uma ação (definição do final do tubo).
- c)Conteúdo é lido a partir de um FIFO (por exemplo: cat /tmp/fifo).

Em outras palavras, se fizermos um script antes do job de backup do Bacula, você pode usar o FIFO para copiar o dump do banco ou o clone da VM sem ocupar espaço no disco rígido do cliente.

Exemplo para o MySQL (ClientRunBeforeJob):

```
#!/bin/sh  
FIFO = "/var/bacula.sql"  
if [-r $ FIFO]; Then  
    rm -f $ FIFO  
    if ["$" -ne "0"]; Then  
        exit 1  
    fi  
fi  
mkfifo $ FIFO  
if ["$" -ne "0"]; Then  
    exit 1  
fi  
screen -m -d -S -t mysqldump su -s / bin / bash -c "mysqldump -u root test -  
p123456> /var/bacula.sql"
```

No exemplo, o comando screen é usado para que o script não seja suspenso e o Bacula fique eternamente à espera da execução. O dump é redirecionado para o FIFO (/var/bacula.sql) e o FileSet deste job deve ser configurado para copiá-lo. Também é necessário usar a opção ReadFifo no FileSet. Exemplo:

```
FileSet {  
    Name = "dump_mysql_fifo"  
    Include {  
        Options {  
            signature = MD5  
            ReadFifo = yes  
        }  
        File = /var/bacula.sql
```



```
}  
}  
}
```

Finalmente, recomenda-se que um script após o job de backup exclua o FIFO criado, já que ele não será mais necessário:

```
#!/ Bin / sh  
FIFO = "/ var / bacula.sql"  
rm -f $ FIFO
```

16.4 Outras considerações sobre o backup de aplicações específicas e exemplos de scripts

16.4.1 Backup de máquinas virtuais

Independentemente do backup em nível de imagem, é aconselhável instalar um cliente Bacula em cada um dos hosts de virtualização, a fim de facilitar o backup e a restauração de nível de arquivo granular. Em outras palavras: tratá-los como se fossem máquinas físicas.

Sobre o backup de nível de imagem, ele poderia ser feito em uma periodicidade menor e serviria mais para fins de recuperação de desastres.

Backup VirtualBox

É muito raro encontrar ambientes de produção de servidores usando o VirtualBox, mas é sempre bom listar opções. O script a seguir executa um VBoxManage nativo antes do job de backup – ClientRunBeforeJob (a palavra-chave clonemedium pode ser diferente de acordo com a versão):

```
VboxManage clonehd "/home/hfaria/VirtualBox VMs/Debian 8/Debian 8.vmdk" /  
root/vbox_bkp/Debian_8_clone.vmdk
```

Neste exemplo, "/home/hfaria/VirtualBox VMs/Debian 8/Debian 8.vmdk" é o caminho para o arquivo de imagem VM, e /root/vbox_bkp/Debian_8_clone.vmdk é o destino da imagem clonada. Essa pasta de destino deve ser incluída no FileSet do Bacula.

Backup de máquinas virtuais Xen

Se não for possível usar bpipe por algum motivo, essas são outras abordagens:

- **Pausando virtual machine Xen enquanto o Bacula efetua a cópia das máquinas virtuais.** A desvantagem deste processo é óbvia: a indisponibilidade temporária da máquina virtual. Se isso não fosse um problema, este seria o seguinte:

- Pause o domU usando o comando "xm pause" (script Bacula ClientRunBeforeJob).
- Sincronize o domU usando o sysrq do xm, ainda no script ClientRunBeforeJob.
- Faça backup das configurações do Xen e do arquivo correspondente à máquina virtual.
- Despouse o "domU" usando "xm unpause".

- **Exportando as máquinas do Xen para disco local e copiando, em seguida, com Bacula.** Desenvolver um script para copiar cada uma das máquinas para um determinado ponto de montagem ou disco com espaço suficiente:

```
snapshot=`xe vm-snapshot uuid=vm_uuid new-name-label=vm_name_snps` && xe  
template-param-set is-a-template=false uuid=$snapshot && xe vm-export
```

```
vm=$snapshot /root/vm_name.xva
```

O Bacula FileSet para o cliente instalado no hypervisor Xen deve incluir a pasta: **/root**.

Também é possível criar um script para buscar todas as máquinas virtuais Xen, como no modelo mostrado em meu site: <<http://www.bacula.com.br/script-configuracao-bpipe-para-backup-de-todas-as-maquinas-virtuais-xen-citrix/>>, e é sempre bom programar a execução após o job de backup (ClientRunAfterJob), para apagar as exportações de VM já copiadas.

Backup de hosts do Microsoft Hyper-v

A Microsoft diz que é possível copiar diretamente pastas e arquivos do Microsoft Hyper-v para um backup seguro. Eles devem incluir o arquivo **InitialStore.xml**, o próprio diretório Hyper-v (**C:\ProgramData\Microsoft\Windows\Hyper-V**) e os volumes contendo os arquivos XML de configuração e VHD(s).

No entanto, alguns componentes devem garantir que esses backups sejam confiáveis. É necessário registrar máquinas virtuais no VSS, Serviço do Windows que permite backup de arquivos abertos, sistema e várias outras coisas.

Ele pode ser feito automaticamente instalando um hotfix (958662) da Microsoft disponível em <<http://support.microsoft.com/kb/958662/pt-br>> ou manualmente através da recomendação do desenvolvedor.

Clique em Iniciar, clique em Executar, digite regedit e, em seguida, clique em OK.

- Localize a seguinte chave do Registro:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion

- Clique com o botão direito do mouse em CurrentVersion, aponte para Novo e, em seguida, clique em Chave.

- Digite **WindowsServerBackup** e, em seguida, pressione ENTER.

- Clique com o botão direito do mouse em **WindowsServerBackup**, aponte para Novo e, em seguida, clique em Chave.

- Digite **Application Support (Suporte a Aplicativos)** e pressione Enter.

- Clique com o botão direito do mouse em **Application Support**, aponte para Novo e, em seguida, clique em Chave.

- Digite **{66841CD4-6DED-4F4B-8F17-FD23F8DDC3DE}** e pressione ENTER.

- Dê um clique direito com o mouse em **{66841CD4-6DED-4F4B-8F17-FD23F8DDC3DE}**, aponte para Novo e, em seguida, clique em **String Value**.

- Digite **Application Identifier** e, em seguida, pressione Enter.

- Clique com o botão direito do mouse em **Application Identifier** e clique em **Modifiy**.

- Na caixa Dados do valor, digite **Hyper-V** e, em seguida, clique em OK.

- No menu File, clique em Sair.

As máquinas virtuais que não têm instalados os serviços de integração serão colocadas em um estado salvo enquanto o backup e o snapshot do VSS (automático do Bacula) forem criados, assim como máquinas virtuais que executam sistemas operacionais que não suportam VSS, como Microsoft Windows 2000 ou Windows XP.

Além disso, observe que as máquinas virtuais que contêm discos dinâmicos devem ser

protegidas off-line ou com um plug-in específico. E o Windows Server Backup não oferece suporte ao backup de máquinas virtuais em volumes compartilhados de cluster do Hyper-V (volumes CSV).

16.4.2 Bancos de dados

As metodologias de backup variam muito de acordo com cada desenvolvedor. Como exemplo, o **PostgreSQL** possui capacidades de backup por **dump** ou backup **quente** (hot backup, através do arquivamento de transações).

Para o **MySQL** você também pode usar dumps, mas a sua técnica de **hot backup** funciona de maneira diferente e é bastante impraticável porque pressupõe fazer uma cópia local dos datafiles (diretório) deste SGBD.

O banco **Oracle** tem semelhanças com o **PostgreSQL**, na medida em que ambos possuem backup PITR quente. No entanto, **Oracle RMAN** é mais elaborado do que **pg_dump**, permitindo mais controle sobre backup e restauração, incluindo backups incrementais.

No caso do **Microsoft SQL**, existe um recurso semelhante à geração de **dump** e **RMAN**, que deve ser integrado ao Bacula através de scripts e gera um backup intermediário em disco para ser copiado pelo Bacula. Após a restauração, o Microsoft SQL fornece uma interface gráfica e de texto e controle sobre a restauração.

PostgreSQL

• pg_dump:

O dump do PostgreSQL ainda é o método mais usado para fazer backup de seus bancos de dados (se você não tem plug-ins específicos para fazê-lo). Um exemplo rudimentar de script ClientRunBeforeJob para gerar o dump (por exemplo):

```
pg_dump dbname > /root/nomebanco.sql
```

Certifique-se de que o destino (ex.: **/root/nomebanco.sql**) esteja incluído no FileSet usado pelo job de backup, para que possa ser copiado pelo Bacula.

• Hot backup:

Em vez do método do dump mais popular, o seguinte método pode ser considerado uma forma mais moderna de fazer backup do PostgreSQL, principalmente porque é um backup mais quente, permite PITR e não requer espaço local para a criação de dump. Muito útil para grandes bases de dados. Converse com seu DBA antes de implementar esse método.

a)Ative o WAL (Write Ahead Log) do PostgreSQL. Dentro do **postgresql.conf**, devem existir as seguintes diretrizes e valores:

```
wal_level = archive
archive_mode = on
archive_command = 'test ! -f /bkp_pgsql/%f && cp %p /bkp_pgsql/%f'
```

b)Crie um diretório para o arquivamento:

```
mkdir /bkp_pgsql
chown postgres /bkp_pgsql
```

Esse caminho armazenará as transações do PostgreSQL e será especialmente útil ao colocar o banco de dados no modo de backup, para que os datafiles possam ser copiados com segurança.

Observação: certifique-se de que o comando `cp -i` funcione para o seu sistema operacional.

c) Crie o script chamado pelo ClientRunBeforeJob. **Por exemplo, `put_pgsql_bkp_mode.sh`:**

```
echo "SELECT pg_start_backup('backup_bacula');" | sudo -u postgres psql
```

d) Crie o script chamado pelo ClientRunAfterJob. Ex.: `exit_pgsql_bkp_mode.sh`:

```
echo "SELECT pg_stop_backup();" | sudo -u postgres psql
```

e) Forneça permissão de execução para ambos os scripts:

```
chmod o+rx put_pgsql_bkp_mode.sh
chmod o+rx exit_pgsql_bkp_mode.sh
```

f) Configure o respectivo job de backup para executar os scripts **Before e After Job**.

g) Inclua no job FileSet o caminho para o diretório dos datafiles do PostgreSQL:

```
...
FileSet = /var/lib/postgresql
```

h) Execute um backup e uma restauração de teste.

MySQL

Os comandos nativos **mysqlhotcopy** ou **mysqldump** podem ser usados para criar um backup confiável a partir dos bancos de dados. O primeiro faz uma cópia a partir do diretório `mysql` datafiles e mantém a disponibilidade durante essa operação, mas nem todos os DBAs preferem usá-la, uma vez que a restauração é menos intuitiva. O método usado por mais de 85% dos administradores do MySQL é o **dump**.

Uma **terceira opção** seria parar o MySQL antes da tarefa de backup através de um script, fazer backup dos arquivos de dados e iniciá-lo novamente após esse backup, mas isso não é abordado neste livro por ser mais rudimentar.

• mysqldump:

Mais uma vez vamos usar o script do Bacula ClientBeforeJob para gerar o dump:

```
mysqldump -u [user] -p[passwd] --all-databases /mnt/mysql_dump/dump.sql
```

...OU...

```
mysqldump -u [user] -p[passwd] --databases database_name /mnt/mysql_dump/
dump.sql
```

... para o dump individual de cada banco de dados.

A palavra-chave **--databases** parece inútil, mas é fundamental para recriar automaticamente bancos de dados desejados ao restaurar o dump.

Certifique-se de que o diretório **/mnt/mysql_dump** existe e está incluído no FileSet de

backup dos servidores.

•mysqlhotbackup:

–Para usar este processo, crie um usuário Bacula no banco de dados a ser feito backup e garanta as permissões para bloquear as tabelas:

```
$ mysql -p Enter password:
```

```
...
```

```
mysql> CREATE USER 'bacula'@'localhost' IDENTIFIED BY 'password';
```

Você receberá a mensagem de sucesso:

```
Query OK, 0 rows affected (0.00 sec)
```

E então:

```
mysql> GRANT SELECT, RELOAD, LOCK TABLES ON *.* TO  
'bacula'@'localhost' IDENTIFIED BY 'password'; Idem: Query  
OK, 0 rows affected (0.00 sec)
```

–Crie um diretório para o destino da cópia quente:

```
mkdir /mnt/mysql_copy
```

–Crie um script shell com o comando “mysqlhotcopy” (ex.: mysql_copy.sh):

```
mysqlhotcopy -regex=. * -user root -password 123456 -allowold /mnt/  
mysql_copy/ > /etc/bacula/mysql_copy.sh
```

–Verificar a permissão para executar o script:

```
chmod o+x /etc/bacula/mysql_copy.sh
```

–Insira a opção ClientRunBeforeJob no job de backup a partir do banco de dados do servidor hospedeiro:

```
Job {  
    Name = "BackupMySQLClient"  
    JobDefs = "DefaultJob"  
    ClientRunBeforeJob = "/etc/bacula/mysql_copy.sh"  
}
```

–Certifique-se de que o caminho do destino da cópia (/mnt/mysql_copy) está incluído no FileSet do job:

```
FileSet {  
    Name = "bd_mysql_set" Include {  
        Options {  
            signature = MD5  
        }  
        File = /mnt/mysql_copy/  
    }  
}
```

–Reinicie ou faça o reload do Bacula Director.

–Submeta um job de backup de teste.

Banco de dados Oracle

Cada abordagem de banco de dados Oracle pode ter suas vantagens e desvantagens. Consulte o DBA de sua empresa para descobrir qual é a mais adequada.

Em **ambos os métodos** é aconselhável definir o Oracle para fazer backups quentes permanentemente:

```
mkdir /backup/oracle_arc  
sqlplus / assysdba
```

```
SQL> shutdown immediate
SQL> LOG_ARCHIVE_DEST='/backup/oracle_arc'
SQL> startup mount
SQL> alter database archivelog;
SQL> alter database open;
```

Integração RMAN:107

Para fins de configuração, é assim que você se conecta ao console de linha de comando RMAN, do shell:

```
rman target /
```

Você pode mostrar e talvez alterar algumas das configurações do RMAN, ou seja, a retenção que será aplicada aos backups intermediários que o RMAN armazenará no disco:

```
RMAN> SHOW ALL;
```

Provavelmente você deseja configurar uma retenção diferente para o backup local do RMAN, uma vez que o Bacula irá copiá-los periodicamente para o armazenamento de backup:

```
RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 3 DAYS;
```

Defina o diretório onde os backups do RMAN serão armazenados:

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/backup/rman/full_%u_%s_%p';
```

Crie esse diretório:

```
mkdir -p /backup/rman
```

Crie um script **ClientRunBeforeJob** para o Bacula executar com esse job de backup do servidor. Deve ser semelhante a isso:

```
echo "BACKUP COMPRESSED AS BACKUPSET DATABASE PLUS ARCHIVELOG;" | "rman
target /"
```

Também é possível fazer backups incrementais RMAN adicionando o seguinte ao comando de backup:

```
INCREMENTAL LEVEL 0
```

Finalmente, garanta que o respectivo FileSet inclua ambos os diretórios:

```
/backup/rman
```

```
/backup/oracle_arc
```

•Oracle hot backup:

a)Configurar o modo de arquivamento, como visto anteriormente.

b)Criar o script ClientRunBeforeJob para o job e colocar o Oracle em modo de backup.[108](#)

Por exemplo: **start-backup-mode.sh** em **/etc/bacula/scripts** com o seguinte:

```
#!/bin/bash
sqlplus /nolog <<EOF
conn sys/manager as sysdba
alter database begin backup; exit
EOF
```

c)Script para sair do modo de backup. Por exemplo: **stop-backup-mode.sh**, na mesma pasta:

```
sqlplus /nolog <<EOF
conn sys/manager as sysdba alter database end backup; exit
EOF
```

d)Execução de scripts com usuário oracle.

O Bacula precisa executar os scripts como usuário do Oracle. Para isso, adicione a seguinte configuração no arquivo **/etc/sudoers** da máquina do cliente:

```
Cmdnd_Alias backup_ORACLE = /bin/su - oracle -c / opt/oraappl/scripts/start-  
backup-mode.sh, /bin/su - oracle -c /opt/oraappl/scripts/start-backup-mode.sh  
bacula LOCAL = NOPASSWD: backup_ORACLE
```

e)Configuração do Bacula. Este é um FileSet provável:

```
FileSet {  
  Name = "Linux-Oracle" Include {  
    Options { signature = SHA1 }  
    File = /mnt/snap/var/oradata  
    File = /mnt/snap/opt/oraappl  
    File = /usr/local/bin  
    File = /var/backup/oracle/arch/  
    File = /etc/oratab  
  }  
}
```

E uma configuração de job provável:

```
Job {  
  Name = "oracle-hot-backup"  
  Client = oracleserver-fd  
  JobDefs = "DefaultJob"  
  RunBeforeJob = "sudo /bin/su - oracle -c /opt/oraappl/ scripts/start-backup-  
mode.sh"  
  RunAfterJob = "sudo /bin/su - oracle -c /opt/oraappl/scripts/stop-backup-  
mode.sh"  
  FileSet = "Linux-Oracle"  
}
```

16.4.3 Servidores web

Microsoft IIS

Um cliente Windows do Bacula será capaz de copiar o conteúdo dos arquivos IIS (Internet Information Services) sem qualquer dificuldade – no entanto, as configurações de servidor da web são omitidas. A Microsoft fornece o script (iisback.vbs), que está bem documentado na ajuda do console de gerenciamento do IIS (executado novamente no ClientRunBeforeJob do Bacula).¹⁰⁹

O script gravará um backup chamado **"baculaiisconfig"** para a pasta: **%systemroot%\System32\inetsrv\MetaBack**:

```
Job {  
  Name = "IISServer"  
  Client = "IIS"  
  FileSet = "IIS"  
  ClientRunBeforeJob = "c:\\WINDOWS\\system32\\iisback.vbs /backup /b  
baculaiisconfig /overwrite /v HIGHEST_VERSION"  
}
```

Apache

O simples backup de pastas do seu site com o Bacula (por exemplo: **/var/www**) seria

suficiente para restaurar o serviço, a menos que haja um aplicativo muito particular em execução. Nesse caso, a documentação de desenvolvimento indicaria procedimentos de backup específicos.

A configuração do Apache e os logs também podem ser facilmente copiados, incluindo **/etc** e **/var** no respectivo FileSet usado pelo job de backup.

16.4.4 Serviços de e-mail

Backup do Microsoft Exchange

A melhor abordagem para o backup do Microsoft Exchange depende de sua própria organização. Como ter apenas um arquivo para armazenar o banco de dados pode retardar o backup, a restauração e eventuais backups diferenciais, é melhor configurar o Exchange para armazenar o banco de dados em vários arquivos menores. De acordo com o desenvolvedor:

*O Exchange suporta vários bancos de dados e grupos de armazenamento no mesmo servidor. Esse suporte permite **dividir um único banco de dados lógico em vários bancos de dados físicos**. Você pode fazer backup e restaurar esses bancos de dados físicos menores muito mais rápido do que bancos de dados maiores. Além disso, você pode ajudar a melhorar a confiabilidade geral do sistema usando vários bancos de dados físicos porque você pode **restaurar um banco de dados individual a partir de um backup** enquanto outros bancos de dados continuam atendendo às solicitações dos clientes.*

***Certifique-se de que os arquivos de banco de dados do Exchange têm nomes de arquivos diferentes.** Ter dois stores com os mesmos nomes de banco de dados causará erros após a restauração, se ambos os stores estiverem sendo restaurados ao mesmo tempo.*

Por exemplo, se o primeiro Storage Group tiver dois armazenamentos de caixa de correio e ambos tiverem arquivos de banco de dados denominados “Priv1.edb” e “Priv1.stm”, esse problema ocorrerá se ambos os stores forem copiados e restaurados ao mesmo tempo.

*Para que a restauração funcione corretamente quando você tem vários armazenamentos no servidor Exchange, **certifique-se de que os arquivos de banco de dados têm nomes diferentes**.*

*Além disso, **certifique-se de que vários bancos de dados no mesmo grupo de armazenamento são nomeados exclusivamente**. Caso contrário, durante um processo de restauração de todo o grupo de armazenamento, os arquivos de patch associados serão gerados com o mesmo nome de arquivo e serão sobrescritos. Isso fará com que a restauração falhe.*

Fazer backup de arquivos que tenham arquivos de banco de dados com os mesmos nomes separadamente (um a um) funcionará.¹¹⁰

Além de dividir o banco de dados do Exchange por tamanho, que pode ser feito para coincidir com cada quota de caixa de correio do usuário, provavelmente também é uma boa ideia criar caixas de correio de arquivo para cada usuário em diferentes bancos de dados também para armazenar e-mails antigos. Esses são mais improváveis de mudar, e, portanto, os backups incrementais não repetiriam as cópias destes.

•Backup do Exchange com o Bacula

O Microsoft Exchange em sua versão 2013 (exemplo) normalmente armazena suas informações de banco de dados neste caminho, que deve ser incluído no backup do servidor

FileSet:
C:\Program Files\Microsoft\Exchange Server\V15\Mailbox\First Storage Group

O Bacula inclui um abrangente suporte ao VSS desde a versão 2.0. Isso deve permitir que você faça backup com segurança de um snapshot consistente do banco de dados do Exchange. Mas o Bacula não envolve os writers VSS de forma sofisticada; por isso, mesmo se você executar um backup completo, isso não será registrado pelo Exchange, e o último timestamp de backup completo não estará definido e os logs de transação não serão liberados. Essa é a forma como ele é construído e não um bug.

Observe que o VSS Writer para Exchange parece estar desabilitado por padrão em algumas versões do Windows Server. Você pode verificar examinando as mensagens VSS, que se parecem com isso:

```
27-Mar 03:45 exchange-fd: VSS Writer (BackupComplete): "System Writer",  
State: 0x1 (VSS_WS_STABLE)  
27-Mar 03:45 exchange-fd: VSS Writer (BackupComplete): "FRS Writer", State:  
0x1 (VSS_WS_STABLE)  
27-Mar 03:45 exchange-fd: VSS Writer (BackupComplete): "MSDEWriter", State:  
0x1 (VSS_WS_STABLE)
```

Observe a falta de uma entrada do Exchange Writer. Esta entrada da base de conhecimento da Microsoft discute como habilitá-lo, o que permitirá que você use pacotes de backup de terceiros, como o Bacula.

Você também pode encontrar o programa utilitário ExMerge. É um aplicativo da Microsoft que permite extrair e mesclar caixas de correio individuais como arquivos PST “de” e “para” um servidor Exchange. Ele deve permitir que você extraia todas as caixas de correio individuais em arquivos PST separados, que seriam então copiados pelo Bacula.¹¹¹

Backup de Cyrus Mail

• Mensagem de e-mail única¹¹²

O Cyrus salva seu e-mail na seguinte pasta, que deve ser copiada pelo Bacula:

```
/var/spool/imap/user/<user_name>/
```

Há um arquivo para cada mensagem. O nome do arquivo consiste em uma série de números e um ponto. Para evitar que as mensagens sejam sobrescritas, você deve criar um diretório adicional para a restauração.

Use a interface de administração da web para criar uma nova pasta dentro do usuário específico na hora de restaurar (por exemplo: .../restore).

```
/var/spool/imap/user/<user_name>/restore/
```

Observação: não crie a pasta manualmente no bash – caso contrário, o Cyrus irá ignorá-la.

Copie as mensagens do backup para o novo diretório. Reinicie a caixa de correio para o Cyrus reconhecer novas mensagens:

```
cyrus@slox:~> reconstruct -r user/<user_name>
```

Ou simplesmente use reconstruct sem opção para redefinir todas as caixas de correio.

•Caixas de e-mail¹¹³

É um procedimento semelhante. Basta restaurar para um subdiretório dentro da caixa de e-mail principal e executar...

```
reconstruct -r -f on the mailbox
```

...para descobrir a caixa restaurada e fornecer um meio de recuperar as mensagens desejadas.

Zimbra Backup

O Zimbra possui um banco de dados interno que precisa de etapas adicionais para um backup seguro. Recomenda-se a utilização de dois jobs de backup diferentes: o primeiro para as mensagens de e-mail e o outro para copiar a pasta da aplicação. Este último job precisa parar o Zimbra antes do início do backup e iniciá-lo após o backup ser concluído.

Exemplo prioridade do **Job1**:

```
Priority=10
```

Exemplo prioridade do **Job1**:

```
/media/storage
```

Exemplo de script ClientRunBeforeJob do **Job2**:

```
vi /opt/bacula/scripts/zimbra_stop.sh
sudo -u zimbra /opt/zimbra/bin/zmcontrol stop
```

Exemplo de script ClientRunAfterJob do **Job2**:

```
vi /opt/bacula/scripts/zimbra_start.sh
sudo -u zimbra /opt/zimbra/bin/zmcontrol start
```

Exemplo prioridade **Job2**:

```
Priority=11
```

Exemplo prioridade **Job2**:

```
/opt/zimbra
```

•Restaurando:

Antes de restaurar, você deve se certificar de que todos os processos associados com a instalação danificada ou falha do Zimbra estejam encerrados. Falhas ao encerrar todos os processos podem ter consequências terríveis.

a) Descubra qual é a versão instalada do Zimbra. Você pode encontrá-la executando os seguintes comandos:

```
cd /opt/zimbra # ou zimbra_broken
cat .install_history
```

Reveja a saída e procure por uma linha perto da parte inferior que mostre a última atualização/instalação que você fez. Parece algo como isso:

```
1203106102: INSTALL SESSION START
1203106124: UPGRADED zimbra-core-4.5.11_GA_1751.RHEL4-20080128130515.i386.rpm
1203106126: UPGRADED zimbra-ldap-4.5.11_GA_1751.RHEL4-20080128130515.i386.rpm
1203106134: UPGRADED zimbra-
logger-4.5.11_GA_1751.RHEL4-20080128130515.i386.rpm
1203106150: UPGRADED zimbra-mta-4.5.11_GA_1751.RHEL4-20080128130515.i386.rpm
1203106154: UPGRADED zimbra-snmp-4.5.11_GA_1751.RHEL4-20080128130515.i386.rpm
```

```
1203106167: UPGRADED zimbra-  
store-4.5.11_GA_1751.RHEL4-20080128130515.i386.rpm  
1203106171: UPGRADED zimbra-  
apache-4.5.11_GA_1751.RHEL4-20080128130515.i386.rpm  
1203106173: UPGRADED zimbra-  
spell-4.5.11_GA_1751.RHEL4-20080128130515.i386.rpm  
1203106173: INSTALL SESSION COMPLETE
```

Para instalações não rpm você pode ver .deb em vez de .rpm, mas isso deve funcionar em todas as plataformas.

b) Renomeie sua instalação “quebrada” do Zimbra.

Você pode ser capaz de limpar dados danificados, se necessário. Mas se você simplesmente não quiser os dados antigos, você pode ignorar essa parte. Exemplo:

```
mv /opt/zimbra /opt/zimbra_broken
```

Você pode querer movê-los completamente para fora da pasta /opt apenas por segurança. Depois disso, restaure o caminho de instalação do Zimbra pelo Bacula.

c) Restaure a pasta **/opt/zimbra** com o Bacula.

d) Faça o download de uma cópia falsa do Zimbra.

Você precisa fazer o download da instalação completa da versão Zimbra que você estava executando anteriormente. Você pode encontrar todas as versões do Zimbra no Sourceforge.

e) Instale a cópia fictícia do Zimbra.

Depois de ter baixado o arquivo tar, você precisará descompactá-lo:

```
tar -xvzf [file_name].tgz
```

Isso criará uma pasta no diretório chamado “zcs”. Abra e execute o script de instalação:

```
cd zcs  
./install.sh
```

Aviso: não execute o script com a opção -u. Você vai remover todos os seus dados da aplicação, e você **precisa** executar o script como root, NÃO zimbra.

Se tudo correr bem, você verá o seguinte:

```
The Zimbra Collaboration Suite appears already to be installed.  
It can be upgraded with no effect on existing accounts,  
or the current installation can be completely removed prior  
to installation for a clean install.  
Do you wish to upgrade? [Y]  
Choose Y.
```

O script removerá os pacotes existentes e os reinstalará. O script tentará parar o Zimbra Services e “UPGRADE FAILED – exiting”. Isso é normal, basta reiniciar o script e ele será instalado normalmente.

Se você receber o erro 389, -1, a conexão recusou erros, pesquise os fóruns. Esses erros são abordados extensivamente.

f) Redefina permissões.

Se estiver funcionando agora, você pode ter mais um obstáculo a superar: permissões.

Talvez seja necessário redefinir as permissões no armazenamento de mensagens executando o seguinte:

```
chown -R zimbra:zimbra /opt/zimbra/store
chown -R zimbra.zimbra /opt/zimbra/index
```

Além disso, execute o seguinte como root:

```
/opt/zimbra/libexec/zmfixperms
```

16.4.5 Serviços de diretório

Para configurações OpenLDAP mais simples que usam somente um único backend, apenas o comando “slapcat” seria suficiente para criar um dump completo para backup. Para cenários mais elaborados (com backends múltiplos), o slapcat precisa do DN (Distinguish Name) de cada um dos backends dos sites – o que também pode ser feito via Bacula com um script antes do job. Um exemplo muito básico de geração do dump:

```
slapcat > /var/ldap.dump
```

Atenção! Seu slapd deve ser **PARADO** durante a execução do **slapcat** (ou pelo menos o modo read-write), sob o risco de comprometer a consistência do banco.

Outra aplicação popular nessa área seria o serviço de diretório 389, que consiste no novo nome do servidor de diretório Fedora/RedHat certificado pela última, embora os dois compartilhem o mesmo código. A metodologia de backup também está na execução do script e na produção de dump através de funcionalidade fornecida pela ferramenta.

¹⁰¹ A largura de banda FIFO depende do sistema operacional, mas geralmente é grande o suficiente para nunca ser considerado um gargalo de desempenho do backup.

¹⁰² Observe que o nome do arquivo é uma figura de idioma, já que o pipe nomeado é considerado um pseudoarquivo.

¹⁰³ bpipe chamará esse programa fazendo um popen nele.

¹⁰⁴ Usando diferentes linhas de comando para bpipe, você pode fazer backup de qualquer tipo de dados (ASCII ou binário), dependendo do programa chamado.

¹⁰⁵ Adaptado de: <http://wiki.bacula.org/doku.php?id=application_specific_backups:postgresql>.

¹⁰⁶ Disponível em: <<https://github.com/lamw/ghettoVCB>>.

¹⁰⁷ Fonte: <<http://www.thegeekstuff.com/2013/08/oracle-rman-backup/>>.

¹⁰⁸ <http://download.oracle.com/docs/cd/B19306_01/server.102/b14231/archredo.htm#i1006184>. <http://download.oracle.com/docs/cd/B28359_01/server.111/b28310/archredo004.htm>.

¹⁰⁹ Fonte: <http://wiki.bacula.org/doku.php?id=application_specific_backups:microsoft_iis>.

¹¹⁰ Referência: <<https://support.microsoft.com/en-us/kb/258243>>.

¹¹¹ Fonte: <http://wiki.bacula.org/doku.php?id=application_specific_backups:exchange_server>.

¹¹² Fonte: <<http://www.novell.com/coolsolutions/qna/1748.html>>.

¹¹³ Fonte: <<http://lists.andrew.cmu.edu/pipermail/info-cyrus/2005-September/019620.html>>.

Procedimentos Específicos do Windows

Atualmente, o Bacula Community Director não está disponível para Windows, mas tanto o daemon de armazenamento como o daemon de arquivo podem ser instalados na plataforma Microsoft. No entanto, o **Storage Daemon para Windows foi relançado recentemente após vários anos de descontinuidade e por isso ainda não está no âmbito deste livro.**

À frente estão a instalação e a operação do Windows Bacula Client para Windows, testado em Windows (9x, NT, 2000, XP, 2003, Windows 7, Windows 8, Windows 8.1, Windows 10, Windows Server 2012/R2, Windows Server 2016). Observe que ainda é possível instalar as interfaces de usuários portadas: BAT e bconsole.

Uma vez instalado, o Bacula roda como um “Windows Service”. Isso significa que ele é inicializado automaticamente, mesmo se não houver nenhum usuário para autenticar.

17.1 Instalando o cliente Bacula e consoles

O download do instalador Bacula para Windows pode ser feito a partir do site do projeto bacula.org <<http://blog.bacula.org/binary-download-center/>>.

Observe que existem arquivos para o Windows **32 ou 64 bits**. Escolha em conformidade.

Proceda com o download do pacote executável (por exemplo: **bacula-win32-7.4.4.exe**).

a) Como administrador, clique duas vezes no arquivo baixado.

b) Iniciada a instalação, clique no botão Next > (**Figura 17.1**) e aceite os termos da licença.



Figura 17.1. Tela inicial de instalação.

c) Sempre escolha **Automatic**, como na **Figura 17.2**, a menos que saiba o que significa a

opção **Custom** (personalizada).

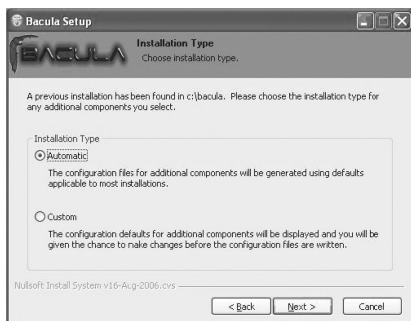


Figura 17.2. Tipo de instalação.

d) Para a instalação do cliente Windows em máquinas de produção (**Figura 17.3**), é aconselhável selecionar apenas a opção **Client**, mas para este manual os Consoles também serão instalados:

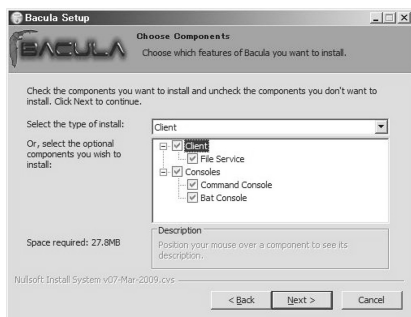


Figura 17.3. Escolher componentes.

e) Na tela seguinte (**Figura 17.4**), você será solicitado a inserir **três valores**. O primeiro só aparece e é necessário para a instalação **do cliente**. Os outros dois **só se aplicam à configuração dos consoles**.

–**Name**: o nome do Director exatamente como configurado no `bacula-dir.conf` do servidor (é sensível a maiúsculas e minúsculas e provavelmente deve incluir o sufixo “-dir”, se estiver presente).

–**Password**: permite que os consoles se conectem ao Director. Ele está localizado em `bacula-dir.conf`, recurso **Director** (primeira senha).

–**Address**: o endereço IP ou FQDN para a máquina que hospeda o Director do Bacula.

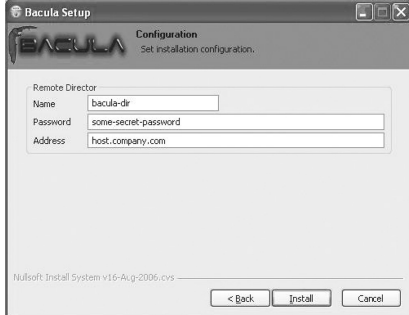


Figura 17.4. Configuração.

f) Após a instalação, seu cliente já deve estar iniciado. Agora acesse a configuração do cliente – **bacula-fd.conf**. Isso é feito através do **menu Iniciar do Windows > Bacula > Configuration > Edit Client Configuration**. É semelhante à configuração de clientes Linux e qualquer modificação requer uma reinicialização do serviço (Windows Service Manager – services.msc).

Observação: se você estiver enfrentando problemas de inicialização do cliente, também poderá iniciá-lo via linha de comando do Windows – cmd:

“C:\Program Files\Bacula\bacula-fd. exe” -c “C:\Program Files\Bacula\bacula-fd.conf”

g) Você precisará buscar dois valores do Windows bacula-fd.conf para anexar este novo cliente ao Director do Bacula. Um é o **File Daemon Name** e outro é o **Director Password** (neste momento é possível alterar essa senha, que é exclusiva para a conexão do Director com esse cliente). Copie e cole as informações em um bloco de notas.

```
...
Director {
    Name = bacula-dir
    Password = "xos2o70KCHf4namn9vs5fENUGNuEf5Wax+Z"
}
FileDaemon {
    Name = windows-fd
    FDport = 9102
}
...
```

h) Agora é hora de configurar o **bacula-dir.conf** (máquina do servidor de backup) com um novo **Recurso Client**, para amarrá-los. Aqui você coloca informações sobre o seu novo cliente Windows:

```
Client {
    Name = windows-fd
    Address = 192.168.0.10
    FDPort = 9102
    Catalog = MyCatalog
    Password = "xos2o70KCHf4namn9vs5fENUGNuEf5Wax+Z"
    File Retention = 60 days
    Job Retention = 6 months
    AutoPrune = yes
}
```

i) Neste momento, reinicie ou recarregue o daemon do Director e teste a conexão com o novo cliente anexado com o comando do **bconsole: status client**. Uma **conexão bem-sucedida** imprimirá a versão do cliente, os jobs em execução e terminados para aquele cliente. O **firewall do Windows** e **erros de configuração** (por exemplo, **Senha** ou **Nome do Director** no bacula-fd.conf do Windows) podem levar a conexões com falha.

j) Agora você pode prosseguir com a configuração usual do job no Bacula. É muito provável que precise de dois novos recursos **Job** e **FileSet** para configurar o backup do novo cliente do Windows.

17.2 System State

Existem vários componentes importantes que não podem ser simplesmente copiados do sistema operacional Windows e precisam de uma etapa anterior para serem armazenados em backup. O System State (estado do sistema) também permite a recuperação de desastres do SO Windows Pitr e também faz parte de outro processo de restauração de aplicativos da Microsoft, como o Active Directory, o Exchange etc.

Os dados de estado do sistema são compostos dos seguintes arquivos:

- *Arquivos de inicialização, incluindo os arquivos do sistema e todos os arquivos protegidos pelo Windows File Protection (WFP).*
- *Active Directory (somente em um controlador de domínio).*
- *Sysvol (somente em um controlador de domínio).*
- *Serviços de certificados (somente na autoridade de certificação).*
- *Banco de dados de cluster (somente em um nó de cluster).*
- *O Registro.*
- *Informações de configuração do contador de desempenho.*
- *Base de dados de registro de classes do Component Services.* [114](#)

a) Para Windows XP e 2000 Server, este é um script de pré-backup de exemplo a ser chamado pelo Bacula para gerar o backup de systemstate (por exemplo: um script **.bat**) [115](#):

```
ntbackup backup systemstate /F c:\systemstate.bkf
```

Para o Windows 2008/2016 Server, ele também requer uma partição separada para armazenar o dump do systemstate, além do OS:

```
wbadmin start systemstatebackup -backuptarget:E: -quiet
```

b) O recurso Job do Bacula para fazer o backup dessa máquina Windows deve conter a opção que chama o script que gera o backup do systemstate...

```
ClientRunClientBeforeJob = C:/systemstate.bat
```

...no respectivo job, no bacula-dir.conf.

c) Por fim, certifique-se de que o **FileSet** inclui o diretório de saída do script de backup gerado pelo sistema (**systemstate.bkf**) ou o volume do **backuptarget** (Windows 2008/2016).

17.3 Exemplos de FileSet de Windows

Esta é uma amostra de FileSet Windows com muitas exclusões de exemplo que são mais úteis para um backup completo da máquina, o que nem sempre é o caso:[116](#)

```
FileSet {
Name = "Windows 2000"

Enable VSS = Yes # Permite o backup de arquivos abertos. Pode ser desabilitado no caso de problemas VSS do Windows.

Include {
  Options {
    signature = MD5
    IgnoreCase = yes
Exclude = yes # Indica que a correspondência de caracteres curinga a seguir será excluída do backup.
# Exclusão de cache do Mozilla:
WildDir = "[A-Z]:/Documents and Settings/*/Application Data/*/Profiles/*/Cache"
WildDir = "[A-Z]:/Documents and Settings/*/ Application Data/*/Profiles/*/Cache.Trash"
WildDir = "[A-Z]:/Documents and Settings/*/ Application Data/*/Profiles/*/ImapMail"

# Exclusão de arquivos do Windows que sempre estão em uso:
WildFile = "[A-Z]:/Documents and Settings/*/Local Settings/Application Data/Microsoft/Windows/userclass.*"
WildFile = "[A-Z]:/Documents and Settings/*/ntuser.*"

# Outras exclusões usuais:
WildDir = "[A-Z]:/Documents and Settings/*/Cookies"
WildDir = "[A-Z]:/Documents and Settings/*/Recent"
WildDir = "[A-Z]:/Documents and Settings/*/Local Settings/History"
WildDir = "[A-Z]:/Documents and Settings/*/Local Settings/Temp"
WildDir = "[A-Z]:/Documents and Settings/*/Local Settings/Temporary Internet Files"

# Arquivos que normalmente não podem ser copiados:
WildFile = "[A-Z]:/Documents and Settings/All Users/Application Data/Microsoft/Network/Downloader/qmgr[01].dat"

# Outros arquivos que podem ser ignorados:
WildFile = "[A-Z]:/WINNT/security/logs/scepol.log"
WildDir = "[A-Z]:/WINNT/system32/config"
WildDir = "[A-Z]:/WINNT/msdownld.tmp"
WildDir = "[A-Z]:/WINNT/Internet Logs"
WildDir = "[A-Z]:/WINNT/$Nt*Uninstall*"
WildDir = "[A-Z]:/WINNT/sysvol"
WildFile = "[A-Z]:/WINNT/cluster/CLUSDB"
WildFile = "[A-Z]:/WINNT/cluster/CLUSDB.LOG"
WildFile = "[A-Z]:/WINNT/NTDS/edb.log"
WildFile = "[A-Z]:/WINNT/NTDS/ntds.dit"
WildFile = "[A-Z]:/WINNT/NTDS/temp.edb"
WildFile = "[A-Z]:/WINNT/ntfrs/jet/log/edb.log"
WildFile = "[A-Z]:/WINNT/ntfrs/jet/ntfrs.jdb"
WildFile = "[A-Z]:/WINNT/ntfrs/jet/temp/tmp.edb"
WildFile = "[A-Z]:/WINNT/system32/CPL.CFG"
WildFile = "[A-Z]:/WINNT/system32/dhcp/dhcp.mdb"
WildFile = "[A-Z]:/WINNT/system32/dhcp/j50.log"
WildFile = "[A-Z]:/WINNT/system32/dhcp/tmp.edb"
WildFile = "[A-Z]:/WINNT/system32/LServer/edb.log"
WildFile = "[A-Z]:/WINNT/system32/LServer/TLSLic.edb"
```

```
WildFile = "[A-Z]:/WINNT/system32/LServer/tmp.edb"
WildFile = "[A-Z]:/WINNT/system32/wins/j50.log"
WildFile = "[A-Z]:/WINNT/system32/wins/wins.mdb"
WildFile = "[A-Z]:/WINNT/system32/wins/winstmp.mdb"
```

Exclusão de arquivos temporários:

```
WildDir = "[A-Z]:/WINNT/Temp"
WildDir = "[A-Z]:/tmp"
```

```
WildFile = "*.tmp" WildDir = "[A-Z]:/tmp"
WildDir = "[A-Z]:/var/tmp"
```

Lixo:

```
WildDir = "[A-Z]:/RECYCLER"
```

Arquivos de "Swap"

```
WildFile = "[A-Z]:/pagefile.sys"
```

```
}
```

Exemplos de Include:

```
File = "C:/" # Barra normal, mesmo para Windows!
File = "D:/Program Files/" # Caminhos com espaços devem estar entre aspas.
}
```

```
}
```

114 Microsoft Technet: <<https://technet.microsoft.com/en-us/library/cc938537.aspx>>.

115 Não existe opção silenciosa para o ntbackup.

116 Disponível em: <http://www.bacula.org/7.4.x-manuals/en/main/Configuring_Director.html>.

Referências

ARAUJO, E. **Noções básicas sobre conceitos de backup**. Rio de Janeiro: Ciência Moderna, 2008.

ASKUBUNTU. **How to check hard disk performance**. Disponível em: <<https://askubuntu.com/questions/87035/how-to-check-hard-disk-performance>>. Acesso em: 02 jun. 2017.

BACULA DO BRASIL. **1 milhão de downloads do Bacula**. 27 jan. 2010. Disponível em: <<http://www.bacula.com.br/1-milhao-de-downloads-do-bacula/>>. Acesso em: 26 maio 2017.

BACULA DOKUWIKI. **Exchange Server**. Disponível em: <http://wiki.bacula.org/doku.php?id=application_specific_backups:exchange_server>. Acesso em: 05 jun. 2017.

BACULA DOKUWIKI. **Microsoft IIS**. Disponível em: <http://wiki.bacula.org/doku.php?id=application_specific_backups:microsoft_iis>. Acesso em: 05 jun. 2017.

BACULA DOKUWIKI. **Oracle**. Disponível em: <http://wiki.bacula.org/doku.php?id=application_specific_backups:oracle>. Acesso em: 26 maio 2017.

BACULA SYSTEMS. **Bacula Administrator II Training Course**. Disponível em: <<https://www.baculasystems.com/bacula-administrator-ii-training-course>>. Acesso em: 31 maio 2017.

BACULA SYSTEMS. Site. Disponível em: <<http://www.baculasystems.com>>. Acesso em: 26 maio 2017.

BACULA WIKI. Site. Disponível em: <<http://wiki.bacula.org/doku.php>>. Acesso em: 26 maio 2017.

BACULA.ORG. **Database Tables**. Disponível em: <http://www.bacula.org/5.1.x-manuals/en/developers/developers/Database_Tables.html>. Acesso em: 26 maio 2017.

BACULA.ORG. **Manuals**. Disponível em: <<http://blog.bacula.org/documentation/documentation/>>. Acesso em: 26 maio 2017.

BACULA.ORG. Site. Disponível em: <<http://www.bacula.org>>. Acesso em: 26 maio 2017.

BOLLENGIER, E.; LEHMANN, A.; SIBBALD, Kern. **Bacula Administration Course II Manual and Slides**, s.d.

BRENNER, B. Zombie machines used in 'brutal' SSH attacks. **TechTarget**, June 02, 2005. Disponível em: <http://searchsecurity.techtarget.com/news/article/0,289142,sid14_gci1094140,00.html>. Acesso em: 26 maio 2017.

CAMÕES, T. **Comparação entre BRF e Métodos Tradicionais para Gerenciamento de backups.** UFBA, 2007.

COUGIAS, D. J.; HEIBERGER, E. L.; KOOP, K; O'CONNELL, L. (ed.) **The Backup Book: disaster recovery from desktop to data center.** Lecanto, FL: Network Frontiers, Schaser-Vartan Books, 2003.

CURTIS, P. **Using SANs and NAS: help for storage administration.** Sebastopol, CA: O'Reilly, 2002.

DIE.NET. **dd(1) – Linux man page.** Disponível em: <<http://linux.die.net/man/1/dd>>. Acesso em: 26 maio 2017.

ESNET. **Linux Tuning.** Disponível em: <<https://fasterdata.es.net/host-tuning/linux/>>. Acesso em: 02 jun. 2017.

FELLER, J.; FITZGERALD, B.; HISSAM, S.; LAKHANI, K. (eds.) **Perspectives on Free and Open Source Software.** Cambridge, MA: MIT Press, 2005.

FELNER, M.; WEICHINGER, S. Data at Risk. **Linux Magazine**, n. 54, p. 53-60.

GARTNER. **O futuro do backup pode não ser backup.** Publicado em 22 set. 2011.

GUISE, P. **Enterprise Systems Backup & Recovery: a corporate insurance policy.** CRC Press, 2008.

HOSTNET. **“Brasil é estrela brilhante no software livre”, afirma Maddog.** 31 jan. 2008. Disponível em: <<https://www.hostnet.com.br/blog/jon-maddog-hall-fala-do-futuro-do-software-a-kaue-linden/>>. Acesso em: 26 maio 2017.

JONES, J. V. **Integrated Logistics Support Handbook.** 3rd. ed. (McGraw-Hill Logistics Series). New York, NY: McGraw-Hill, 2006.

MEFFE, C. **O software público e suas qualidades extrínsecas.** Disponível em: <https://softwarepublico.gov.br/social/articles/0003/7467/Software_P_blico_e_a_suas_qualidades_extr_nsecas.pdf>. Acesso em: 26 maio 2017.

MELLOR, C. LTO issues a mighty seventh-generation 15TB tape format. **The Register**, Sep. 16, 2015. Disponível em: <http://www.theregister.co.uk/2015/09/16/lto_has_15tb_gen_7_tape_format/>. Acesso em: 30 maio 2017.

MICROFOCUS. **Restoring email saved in Cyrus.** Apr. 30, 2004. Disponível em: <<http://www.novell.com/coolsolutions/qna/1748.html>>. Acesso em: 05 jun. 2017.

MICROSOFT TECHNET. **Backing Up System State Data.** Disponível em: <<https://technet.microsoft.com/en-us/library/cc938537.aspx>>. Acesso em: 05 jun. 2017.

MICROSOFT. **How to back up and restore an Exchange computer by using the Windows Backup program.** Disponível em:<<https://support.microsoft.com/en-us/kb/258243>>. Acesso em: 05 jun. 2017.

MINASI, M. **Dominando o Windows 2000 Server.** São Paulo: M. Books, 2012.

MINISTÉRIO DO MEIO AMBIENTE. **O que é Interoperabilidade?**. Disponível em: <<http://www.mma.gov.br/governanca-ambiental/geoprocessamento/item/869-interoperabilidade-o-que-%C3%A9>>. Acesso em: 26 maio 2017.

MITCHELL, D. Computer Associates ARCserve Backup 12.5 review. **Alphr**, May 27, 2009. Disponível em: <<http://www.alphr.com/computer-associates/computer-associates-arcserve-backup-125/30172/computer-associates-arcserve>>. Acesso em: 26 maio 2017.

NATARAJAN, R. How to Backup Oracle Database using RMAN (with Examples). **The Geek Stuff**, Aug. 27, 2013. Disponível em: <<http://www.thegeekstuff.com/2013/08/oracle-rman-backup/>>. Acesso em: 05 jun. 2016.

OLIVA, A.; TAURION, C.; ANDERSON, C.; SENA, J.; FERREIRA, M.; MICHELAZZO, P.; REZENDE, P.; MIZUKAMI, P.; QUEIROZ, R.; MELO, T. **A Revolução do Software Livre**. Comunidade Sol, 2009.

PANKOV, E. **Network Backup with Bacula [How-to]**. Birmingham, UK: Packt Publishing, 2012.

PETRELEY, N. Security Report: Windows vs Linux. **The Register**, Oct. 22, 2004. Disponível em: <https://www.theregister.co.uk/2004/10/22/security_report_windows_vs_linux/>. Acesso em: 26 maio 2017.

POSTGRESQL. **Vacuumdb**. Disponível em: <<https://www.postgresql.org/docs/9.0/static/app-vacuumdb.html>>. Acesso em: 02 jun. 2017.

RIBEIRO, U. **Certificação Linux**. Rio de Janeiro: Axcel Books, 2004.

SIBBALD, K. **Bacula Main Reference**: the leading open source backup solution. Disponível em: <<http://www.bacula.com.br/manual/>>. Acesso em: 31 maio 2017.

SIBBALD, K. **Bacula Problem Resolution Guide**: the leading open source backup solution. Aug. 14, 2015. Disponível em: <http://www.bacula.org/7.2.x-manuals/en/problems/Problem_Resolution_Guide.html>. Acesso em: 26 maio 2017.

SMANIOTO, C. E. Otimização: uma ferramenta chamada vacuum. **Devmedia**, s.d. Disponível em: <<http://www.devmedia.com.br/otimizacao-uma-ferramenta-chamada-vacuum/1710>>. Acesso em: 02 jun. 2017.

STACKOVERFLOW. **What is a simple command line program or script to backup SQL server databases?** Sep. 23, 2008. Disponível em: <<http://stackoverflow.com/questions/122690/what-is-a-simple-command-line-program-or-script-to-backup-sql-server-databases>>. Acesso em: 26 maio 2017.

SUPERUSER. **SSD 2 million hour MTBF? How is this proven?** Aug. 4, 2015. Disponível em: <<https://superuser.com/questions/950825/ssd-2-million-hour-mtbf-how-is-this-proven>>. Acesso em: 30 maio 2017.

TANENBAUM, A. **Redes de Computadores**. 4.ed. Rio de Janeiro: Campus, 2003.

TAPSCOTT, D.; WILLIAMS, A. **Wikinomics**: como a colaboração em massa pode mudar o seu negócio. Rio de Janeiro: Nova Fronteira, 2007.

TOM'S HARDWARE. **[05] Write Throughput Maximum: h2benchw 3.16**. Disponível em: <<http://www.tomshardware.com/charts/hdd-charts-2013/-05-Write-Throughput-Maximum-h2benchw-3.16,2903.html>>. Acesso em: 30 maio 2017.

USERBENCHMARK. Site. Disponível em: <<http://ssd.userbenchmark.com/>>. Acesso em: 30 maio 2017.

VMWARE. **VMware Consolidated Backup 1.0.3 Release Notes**. July 31, 2007. Disponível em: <http://www.vmware.com/support/vi3/doc/releasenotes_vcb103.html>. Acesso em: 26 maio 2017.

WIKIPEDIA. **Backup**. Disponível em: <<http://en.wikipedia.org/wiki/backup>>. Acesso em: 26 maio 2017.

WIKIPEDIA. **GFS**. Disponível em: <<http://en.wikipedia.org/wiki/GFS>>. Acesso em: 26 maio 2017.

WIKIPEDIA. **Tolerância a falhas (hardware)**. Disponível em: <[https://pt.wikipedia.org/wiki/Toler%C3%A2ncia_a_falhas_\(hardware\)](https://pt.wikipedia.org/wiki/Toler%C3%A2ncia_a_falhas_(hardware))>. Acesso em: 30 maio 2017.

ZHENG, H. **Tutorial on how to backup/restore cyrus**. Sep. 19, 2005. Disponível em: <<https://lists.andrew.cmu.edu/pipermail/info-cyrus/2005-September/019620.html>>. Acesso em: 05 jun. 2017.

ZIMBRA. Site. Disponível em: <http://wiki.zimbra.com/index.php?title=Bacula_pre/post_cold_backup_scripts>. Acesso em: 26 maio 2017.

Anexo 1.

Página de “Cola” do Bacula (bconsole)

O que está acontecendo?¹¹⁷

Que arquivos são copiados pelos backups?
Que o servidor está fazendo?
Qual é o status de um determinado job?
Que o client está fazendo?
Que o storage está fazendo?
Existe alguma nova informação?

Fazendo backup

Como fazer um backup executar.
Como executar um (ou mais) fitas.

Restaurando

A maneira mais comum (um usuário acidentalmente removeu um arquivo e quer a versão mais recente de volta das fitas):

- Use o comando *restore*.
- Escolha a opção 5 (“**Seleção do backup mais recente para um cliente**”).
- `cd/ls/dir/mark/markdir/unmark/unmarkdir/lsmark/estimate/pwd/count/find`
- `done`

Jobs¹¹⁸

Usar jobs para um job específico
Que jobs estão no storage por cada job
Que jobs são backupados?

Fitas

Que fitas estão nas pools?
Que fitas estão em mídia?
Que pools estão definidas?
Que fitas estão sendo usadas por um certo job?
Qual é o nome dos parâmetros de uma fita

Reutilizar fitas gravadas com outro software

Apagar o conteúdo de uma fita (nd shell)

Anexo 2.

Códigos de Jobs do Bacula

Esses códigos são mostrados pelo comando `list jobs do bconsole` e em algumas interfaces gráficas para listar jobs em execução e terminados. São três categorias de códigos:

- Código do nível de backup
- Código do tipo de job
- Código do status do job

E eles significam:

Significado do código do nível de backup	
Backup full (completo)	F
Backup incremental	I
Backup diferencial	D
Verificação do catálogo	C
Verificação do init db	V
Verificação do volume em face do catálogo	O
Verificação do disco em face do catálogo	d
Verificação dos dados no volume	A
Job de base	B
Restore or admin job	???

Significado	Código do
Backup	B
Verificação	V
Restauração	R
Controle de job de cópia	c
Controle de job de migração	g
Job que foi migrado (e provavelmente tem dados descartáveis)	M
Job de cópia	C
Interno do sistema	I
Admin (tarefas de administração)	D
Archive (arquivo)	A
Scan	S

Significado	Código do s
Cancelado pelo usuário	A
Bloqueado	B
Criado, mas não está em execução	C
Esperando pelo recurso cliente	c
Verificando diferenças	D
Esperando a quantidade máxima de jobs	d
Terminado em erro	E

Erro não fatal	e
Erro fatal	f
Esperando pelo cliente	F
Esperando pelo recurso do job	j
Esperando por um mount	M
Esperando nova mídia	m
Esperando jobs de maior prioridade terminarem	p
Em execução	R
Scan	S
Esperando pelo recurso storage	s
Terminado normalmente	T
Esperando o horário de início	t

Anexo 3.

Expansão de Variáveis

Funcionalidade geral¹¹⁹

Esta é basicamente uma capacidade de expansão de string que permite referenciar variáveis, matrizes de indexação, substituição condicional de variáveis, conversão de casos, seleção de substring, correspondência de expressão regular e substituição, substituição de classe de caracteres, strings de preenchimento, expansão repetida em um loop controlado pelo usuário, suporte a expressões aritméticas no início do loop, condições de passo e final e expansão recursiva.

Ao usar caracteres de expansão variável (por exemplo) para formar nomes de volumes (diretiva Label Format, recurso Pool, bacula-dir.conf), o valor deve sempre ser colocado entre aspas (“”).

Por exemplo, \${HOME} será substituído pelo diretório pessoal conforme definido no ambiente do sistema. Se você definiu a variável xxx para ser **Test**, então a referência \${xxx:p/7/Y/r} irá direto para o conteúdo de xxx com um comprimento de sete caracteres preenchendo com o caractere Y, resultando em **YYYTest**.

Variáveis do Bacula

Dentro do Bacula, existem três classes principais de variáveis com algumas pequenas variações dentro das classes. As classes são definidas adiante.

Contadores

Os contadores são definidos pelo recurso Counter no arquivo bacula-dir.conf. O contador pode ser um contador temporário, que permanece durante a execução do Bacula, ou pode ser uma variável que é armazenada no catálogo e assim retém seu valor de uma execução do Bacula para outra. As variáveis do contador podem ser incrementadas com um sinal de mais (+ após o nome da variável).

Variáveis internas

As variáveis internas são somente leitura e podem estar relacionadas ao job atual (por exemplo, nome do job) ou a variáveis gerais, como a data e a hora. Estão disponíveis as seguintes variáveis:

- Year: o ano atual completo.
- Month: o mês atual (1-12).

- Day: o dia do mês (1-31).
- Hour: a hora (0-24).
- Minute: o minuto atual (0-59).
- Second: o segundo atual (0-59).
- WeekDay: o dia atual da semana (0-6 com 0 sendo domingo).
- Job: o nome do job.
- Dir: o nome do Director.
- Level: o nível do job.
- Type: o tipo do job.
- JobId: o JobId.
- JobName: o nome de trabalho único composto de job e data.
- Storage: o nome do daemon de storage.
- Client: o nome do cliente.
- NumVols: o número atual de volumes na pool.
- Pool: o nome da pool.
- Catalog: o nome do catálogo.
- MediaType: o tipo de mídia.

Variáveis de ambiente

As variáveis de ambiente são somente leitura e devem ser definidas no ambiente antes da execução do Bacula. As variáveis de ambiente podem ser escalares ou uma matriz (array), onde os elementos da matriz são referenciados subscrevendo o nome da variável (por exemplo, `${Months[3]}`). As matrizes de variáveis de ambiente são definidas pela separação dos elementos com um pipe (`|`), portanto, `Months="Jan|Feb|Mar|Apr|..."` define uma variável de ambiente chamada `Months` que será tratada como uma matriz e a referência `${Months[3]}` resultará em `Mar`. Os elementos da matriz podem ter comprimentos diferentes.

Sintaxe completa

Uma vez que a sintaxe é bastante extensa, a seguir você encontrará a pseudo BNF¹²⁰ (metassintaxe). Os caracteres especiais têm o seguinte significado:

```
definição
agrupando se os parênteses não estão entre aspas
$para alternativas
literal / (ou qualquer outro caractere)
na sequência de caracteres ou caracteres
item anterior pode ser repetido zero ou mais vezes
item anterior pode aparecer zero ou uma vez
item anterior deve aparecer uma ou mais vezes
```

E o pseudo BNF descrevendo a sintaxe é:

```
input ::= ( TEXT
```

Semântica

Os itens listados no comando anterior, que sempre seguem dois pontos (:), têm os seguintes

significados:

- executar a substituição se a variável estiver vazia
- + executar substituição se a variável não estiver vazia
- o cortar substring do valor da variável
- # comprimento do valor da variável
- * substitui string vazia se o valor da variável não estiver vazio, caso contrário, substitui o parâmetro que segue
- s pesquisa de expressão regular e substitui. As opções de casamento são: m = multilinha, i = não diferencia maiúsculas de minúsculas, g = global, t = texto plano (sem regexp)
- y transpor caracteres da classe A para a classe B
- p pad variável para l = esquerda, r = direita ou c = centro, com segundo valor
- % chamada de função especial (nenhuma implementada)
- l torna minúscula o valor da variável
- u torna maiúscula o valor da variável

Os valores de loop_limits (limites de loop) são: start, step e end.

Um nome de variável de contador seguido imediatamente por um sinal de adição (+) fará com que o contador seja incrementado em 1.

Exemplos

Para criar uma data ISO:

DLT-\${Year}-\${Month:p/2/0/r}-\${Day:p/2/0/r}

Em 20 de junho de 2003, forneceria: **DLT-2003-06-20**.

Se você definir a variável de ambiente **mon**:

January|February|March|April|May|...

File-\${mon[\${Month}]}/\${Day}/\${Year}

Em **1º de março**, daria: **File-March/1/2003**

Licenças do Bacula Enterprise, Treinamento e Suporte Profissionais do Bacula Community

Heitor Faria é CEO da **Bacula do Brasil**, distribuidora exclusiva do Bacula Enterprise no Brasil. Além disso, é fundador e instrutor da Comunidade Nacional do Bacula.

Com mais de duzentas aulas e apresentações em todo o mundo, ministra treinamentos nas modalidades **in-company** e através de **videoaulas**.

Para mais informações:

- Conheça e compre o Bacula Enterprise:
<<http://www.bacula.com.br/enterprise/>>.
- Adquira treinamento in-company:
<<http://www.bacula.com.br/in-company/>>.
- Compre as videoaulas:
<<http://www.bacula.com.br/treinamento-bacula-ed/>>.
- Para contato:
<<http://www.bacula.com.br/fale-conosco-geral/>>.

¹¹⁷ Fonte: <<https://workaround.org/bacula-cheatsheet>>.

¹¹⁸ Verifique o próximo anexo para conhecer os códigos dos jobs.

¹¹⁹ Adaptado e traduzido do Manual do Bacula: <http://www.bacula.org/7.4.x-manuals/en/misc/Variable_Expansion.html>.

¹²⁰ BNF, do inglês Backus-Naur Form ou Backus Normal Form. Referência: <https://pt.wikipedia.org/wiki/Formalismo_de_Backus-Naur>.